DEFENSE INFORMATION SYSTEMS AGENCY

WEB SERVICE STANDARD

C2 USER REQUIREMENTS

FINAL

8 DECEMBER 2003

Table of Contents

1 Executive Summary

1.1 Related DoD Guidance

The DoD is clearly already on a path to full adoption of Web Services. The DoD Transformation Planning Guidance defines an environment where future information technology solutions will require greater interoperability, broader use of wide area network resources, and more rapid integration of information. These and other guidance criteria are naturally met by web services technologies and the products that are currently emerging to support these standards. The Joint Technical Architecture (JTA) already recommends basic Web Services technologies. We recommend the JTA be updated regularly to accommodate additional and future web services standards that will provide opportunities for more interoperability and more productivity in the integration and/or development of DoD systems.

1.2 C2-Related Initiatives

The Horizontal Fusion (HF) Portfolio Initiative was created early in 2003 to reach across traditionally stove-piped organizations, and integration and aggregation of information through net-centric means. Through Net-Centricity, users are able to seek the information they need across the battlespace through smart pull and through information sharing channels. The ultimate goal of HF is to allow discovery of and access to the right information at the right time by the right people regardless of mission. Web services are a logical implementation mechanism for these goals because of the ubiquity of HTTP and XML, and now SOAP and WSDL.

GIG Bandwidth Expansion (GIG-BE), part of the larger Transformational Communications concept, will supply important networking resources, and a 100 fold increase in bandwidth to major sites around the world. System architects considering web services should take these new resources into account when planning for the future, but should also realize that highly mobile environments will still suffer bandwidth degradation, which could be significant in many cases.

Network Centric Enterprise Services (NCES) is a program for providing enterprise services in support of the Global Information Grid (GIG). These GIG enterprise services (GES) are divided into two categories, community of interest (CoI) services, and core enterprise services (CES). While NCES is still being defined, Web Service technology has been identified as a key component of the early generations of the architecture. NCES is also expected to leverage COTS technologies and products extensively and it is believed that Web Services standards will help support this goal.

JC2 will be implemented on top of NCES and use its services to implement net-centric C2 and related capabilities (through Mission Capability Packages—MCPs). JC2 is expected to be the follow-on program to GCCS, as GCCS capabilities migrate to the network. Transformation of existing GCCS components holds potential to expedite the creation of JC2, but it is not without its difficulties. Many tools may be used to assist in wrapping existing GCCS services in JCS compliant services, but not all of them will be equal in their abilities to provide scalability and availability. As discoveries related to individual tools and software packages are made it is critical that these discoveries by made generally available among the entities tasked with migrating GCCS services to JC2.

For the Navy, concepts like FORCEnet and initiatives supporting FORCEnet implementations, such as Reusable Application Integration and Development Standards (RAPIDS) are emerging to improve the use of network resources as they continue to expand. RAPIDS emphasizes the needs of sophisticated fleet users to extend existing applications by mixing and matching components from various applications, and building systems from a small number of existing components that are downloadable over the network or connecting to services that are hosted by a third party. Part of RAPIDS is to provide guidance to developers to provide specific architectural direction so that they are able to construct their applications into web services that can be easily adopted into a web portal; a basic, web native application; or Java environment with little or no re-coding.

The Navy also started the Task Force Web initiative, to provide the C2 user the ability to access multiple web applications through a single point of entry (Single Sign-On), in a net-centric architecture. SAML and federated identity services (such as the Liberty Alliance or Microsoft Passport) should be investigated as a future evolution of DoD Single Sign-On implementations for web-based applications. TFW also promotes a network-centric computing approach wherein a C2 user can login to any desktop with a browser and access web applications, and web portal user interfaces. We recommend TFW, or its follow-on programs, adopt the WSRP standard to support better web portal interoperability.

Lastly, the DoD Metadata Registry and Clearinghouse is currently storing XML tag and XML Schema definitions. For Web services, it could serve as a stable repository for other kinds of data that is at the same level of abstraction as XML Schema documents. An important example of this other kind of data would be WSDL documents, which themselves include XML Schema documents (which may in fact be registered in the DoD Metadata Registry and Clearinghouse). Another example of similar data that could be registered is taxonomies and ontologies encoded in OWL.

1.3 Security

We recommend further research and implementation of Web Services security solutions. SAML, XML Signature, XML Encryption, and related standards (outlined in this document and described in the Web Services Standards Analysis Report) should be investigated and a strategy for implementation and even DoD-wide standardization should be created. Systems and net-centric capabilities with true security will only exist when the developers and integrators are given specific guidance and well-defined solutions, by standing up security services on DoD networks, and providing software development kits and examples of their use.

1.4 Messaging

C2 systems historically make extensive use of messaging, and much of this messaging has been based on very DoD-specific standards for text and binary representation. These specifications don’t lend themselves well to interoperability with commercial products. A movement toward Web Services standards, such as XML and HTTP (for example) would create more opportunity for reuse, and for leveraging Commercial Off-The-Shelf

(COTS) tools. In addition, these formats would be understood by a large base of software developers, which reduces maintenance and lifecycle costs. The DoD should continue ongoing efforts to create XML Schema-based message standards to eventually supersede the legacy formats.

In addition to HTTP, reliable messaging specifications (e.g. WS-Reliability, ebMS, WS-ReliableMessaging, and WS-Acknowledgement), which are still under development, should be tracked. In the mean time, if reliable messaging is important, selecting a JMS-compliant MOM vendor may be a good option.

1.5 OGC Standards

For Geospatial applications of Web Services, we recommend the DoD support and recommend the use of Open GIS Consortium (OGC) standards, and sponsor ongoing efforts to improve and expand the existing standards:

- Since all OGC standards use well-defined protocols and formats, the user can potentially see many types of data on the same map that were previously visible only separately.
- Because server interfaces are carefully defined, new servers can be deployed without virtually no impact on the other existing servers and clients.

Development time is decreased for new clients and servers since a large body of freely available open-source code exists for implementing OGC services and clients.

These standards have been applied in existing C2 applications (many of them described elsewhere in this document). We recommend the DoD officially adopt these OGC standards as the basis for implementation of web-based geospatial services, and the basis for XML representation of geospatial data.

1.6 CJMTK

CJMTK is a COTS product with proprietary APIs. However, using special connectors, data within ArcIMS can be published in a form compliant with the OGC specifications for Web Map Servers (WMS) and Web Feature Servers (WFS) so that clients already written to these specifications have access to data published by ArcIMS. This significantly broadens the potential client base and integration possibilities for ArcIMS. The DoD should require the use of the OGC standards for all web services access to CJMTK, and the CJMTK acquisition authority should ensure that these standards are properly implemented, maintained, and advanced as new versions emerge.

1.7 Symbology

C2 systems make extensive use of symbology, and MIL-STD-2525B is the mandated DoD standard, however in this new age of net-centric computing, a standard for XML representation of MIL-STD-2525B does not yet exist. We recommend that the DoD define an XML adaptation of MIL-STD-2525B that is based on commercial standards such as the OGC standards Geographic Markup Language (GML) and Styled Layer Descriptor (SLD). A standard schema for symbology would be critical to supporting net-centric C2 computing architectures and system interoperability.

1.8 Web Portals, WSRP, and JSR-168

C2 systems are migrating to web browser user interfaces, and C2 users often prefer a web portal environment to help them manage their web based resources. Web "portlets" can be created as "components", which C2 users can selectively load into their portal framework. Having a large set of these components readily available and exposing a wide variety of data sources allows users to quickly adapt to changing conditions. The leveraging of JSR-168 and WSRP, and the component market they are likely to create holds potential to significantly speed development and reduce development cost.

1.9 Case Studies

In this document, we also explore a few case studies—existing C2 systems that are already leveraging web services including:

- XTCF (Extensible Tactical C4I Framework) – using SOAP, WSDL, and UDDI for interoperable messaging between C2 data management components, but which also uses binary messaging over MOM products as an alternative when greater performance and reliability is desired. Applications like these help to validate the appropriate and inappropriate uses of web services technologies, and also help to drive requirements for better web services standards in the future, such as the need for standardized XML compression.
- GCSS Web Portal – This application integrates web based user interfaces with web services to drive the content of those user interfaces. The use of SOAP and WSDL will become mainstream as a way to expose common services to a wider variety of applications. Additional work is needed in the area of security to ensure only approved access to the services and data. This effort also demonstrated the use of COTS techniques to quickly expose J2EE EJB operations as SOAP web services.
- CFn (Composable FORCEnet) – using OGC Web Feature Server and Web Map Server standards, plus a custom WFS Notification implementation. This effort demonstrates a valid and useful way of integrating information from a wide variety of sources into an integrated picture for the end user. It also points out the need for a standard for WFS Notification, which is lacking in the OGC specifications. We recommend the DoD sponsor an OGC initiative to create a WFS notification standard that meets the requirements of applications like CFn.

2 Introduction

This report presents recommendations on how Web Services Standards should be applied to the Department of Defense (DoD) Command and Control community. It is the second in a series of papers that also includes:

- Analysis of Web Services Standards
- Emerging Web Services Development Environment

This overall effort involves analysis of existing and emerging (proposed) standards supporting Web Services, and evaluates the potential impact on DoD Command and Control (C2), in order to:

(1) Influence use of commercial standards to promote DoD interests

(2) Develop and convey an understanding of Web Services standards issues from a variety of Web Services standard organizations; and

(3) Disseminate timely information concerning commercial standards to DoD users.

This report references the standards discussed in the Analysis of Web Services Standards and recommends Web Services technologies, standards, and practices that should be applied in the evolutional development of C2 and related DoD systems.

3 Related Guidance

Web Services technology is based on web technologies that have been in use for several years, and the DoD has already put in place guidance that impacts the use of web services. This section discusses the related DoD guidance, and recommendations for changes to that guidance, if necessary.

3.1 Department of Defense Transformation Planning Guidance

[Source: *Transformation Planning Guidance, April 2003*, Department of Defense Office of Force Transformation (DoD OFT) http://www.oft.osd.mil .]

The *Transformation Planning Guidance* (TPG) outlines the DoD initiative to transition the military to the information age. The initiative is broad-based, covering technological aspects as well as transformational changes to the organization, business processes, doctrine, strategy, and operational concepts of the DoD. A major thrust of this guidance is toward development of *joint* (i.e. a combination of two or more military Services) interoperability.

The guidance approaches technology broadly, and does not discuss particular implementations, methods, or standards. The guidance is useful in providing scope to other subordinate initiatives, for example the Joint Technical Architecture (JTA) and Coalition interoperability.

The areas of the guidance that touch on or are otherwise relevant to Web Services Standards consist of general direction to develop, promote, and adopt the following:

1. Technical solutions to assist integration of national power. This includes those capabilities that enhance coordination among federal agencies, as well as across all levels of government (federal, state, and local).
2. Technology that increases information sharing, across defense branches and with coalition partners, through a secure network that provides actionable information at all levels of command.
3. Technology that assures information systems in the face of attack.
4. Technology used in conducting offensive information operations.
5. An interoperable joint Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) architecture capability that includes a tailorable joint operational picture.
6. Deployable joint command and control systems for the Standing Joint Force Headquarters.
7. A common relevant operational picture for joint forces.
8. Enhanced intelligence, surveillance, and reconnaissance capabilities.
9. Selected sensor-to-shooter linkages prioritized by contribution to the joint operation.
10. Reachback capabilities that provide global information access.

11. Adaptive mission planning, rehearsal, and joint training linked with C4ISR.

3.2 Joint Technical Architecture

[Source: *Department of Defense Joint Technical Architecture v5.0*, 4 April 2003.]

The DoD Joint Technical Architecture (JTA) provides the minimum set of standards that, when implemented, facilitates seamless flow of information among all military tactical, strategic, and supporting elements. The JTA is documented in the *Department of Defense Joint Technical Architecture*.

The JTA consists of both mandated and emerging standards. Mandated standards are stable, technically mature, and publicly available. Emerging standards are those that do not yet meet these guidelines, but are expected to within the next three years.

The JTA architecture consists of the JTA Core and the JTA domains. The JTA Core contains the minimum set of JTA elements applicable to all DoD systems to support interoperability.

The JTA Core consists of the following:

- Information processing. This includes Government and commercial information processing standards the DoD uses to develop integrated, interoperable systems.

  The document interchange service specifies the supported data structures to be used for storage of electronic information and its transmission between information systems.

  Mandated Document Interchange standards relevant to Web Services are:

  - *ISO 8879:1986, Information processing Text and office systems Standard Generalized Markup Language (SGML) with Amendment 1*, 1988, *Technical Corrigendum 1*:1996 and *Technical Corrigendum 2*:1999.
  - *HTML 4.01 Specification, W3C Recommendation*, 24 December 1999.
  - *Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation*, 6 October 2000.
  - *XML Schema Part 1: Structures, W3C Recommendation*, 2 May 2001.
  - *XML Schema Part 2: Datatypes, W3C Recommendation*, 2 May 2001.
  - *Namespaces in XML, W3C Recommendation*, 14 January 1999.

  Emerging Document Interchange standards relevant to Web Services are:

  - *XHTML" 1.0: The Extensible HyperText Markup Language, Second Edition, A Reformulation of HTML 4 in XML 1.0, W3C Recommendation*, 26 January 2000, revised 1 August 2002.
  - *XForms 1.0, W3C Working Draft*, 12 November 2002.
  - *XForms Requirements, W3C Working Draft*, 4 April 2001.
  - *Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation*, 22 February 1999, REC-rdf-syntax-19990222.

- *Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation*, 27 March 2000, CR-rdf-schema-20000327.
- *Extensible Stylesheet Language (XSL), Version 1.0, W3C Recommendation*, 15 October 2001.
- *XSL Transformations (XSLT), Version 1.1, W3C Working Draft*, 24 August 2001.
- *XML Path Language (XPATH), Version 1.0, W3C Recommendation*, 16 November 1999.
- *XML-Signature Syntax and Processing, W3C Recommendation*, 12 February 2002.
- *XQuery 1.0, An XML Query Language, W3C Working Draft*, 15 November 2002.
- *Web Services Description Language (WSDL) 1.1, W3C Note*, 15 March 2001.
- *Simple Object Access Protocol (SOAP) 1.1, W3C Note*, 08 May 2000.
- *UDDI Version 3.0 Published Specification*, 19 July 2002.
- *Cascading Style Sheets (CSS) Level 1 (CSS1), W3C Recommendation*, 17 December 1996.
- *Document Object Model (DOM) Level 1 Specification, Version 1.0, W3C Recommendation*, 1 October 1998.

Raster Product Format (RPF) defines a common format for the interchange of raster-formatted digital geospatial data among DoD Components. Existing geospatial products that implement RPF include Compressed ARC Digitized Raster Graphics (CADRG), Controlled Image Base (CIB), and Digital Point Positioning Data Base (DPPDB).

For raster-based products, the following standard is mandated:

- *MIL-STD-2411, Raster Product Format*, 6 October 1994; with *Notice of Change, Notice 1*, 17 January 1995, and *Notice of Change, Notice 2*, 16 August 2001.

Vector Product Format (VPF) defines a common format, structure, and organization for data objects in large geographic databases based on a georelational data model and intended for direct use. Existing geospatial products that implement VPF include: Vector Map (VMap) Levels 0-2, Urban Vector Map (UVMap), Digital Nautical Chart (DNC), VPF Interim Terrain Data (VITD), Digital Topographic Data (DTOP), and World Vector Shoreline Plus (WVSPLUS).

For vector-based products, the following standard is mandated:

- *MIL-STD-2407, Interface Standard for Vector Product Format (VPF)*, 28 June 1996, with *Notice of Change, Notice 1*, 26 October 1999.

World Geodetic System (WGS 84), a Conventional Terrestrial Reference System (CTRS), is mandated for representation of a reference frame, reference ellipsoid, fundamental constants, and an Earth Gravitational Model with related geoid.

WGS 84 will be used for all joint operations and is recommended for use in multinational and unilateral operations after coordination with allied commands. The following standard is mandated:

- *MIL-STD-2401, Department of Defense Standard Practice, World Geodetic System (WGS)*, 11 January 1994, as implemented by *NIMA TR 8350.2, Department of Defense World Geodetic System 1984: Its Definitions and Relationships with Local Geodetic Systems, Third Edition*, 4 July 1997, as modified by *Amendment 1*, 3 January 2000.

FIPS PUB 10-4 provides a list of the basic geopolitical entities in the world, together with the principal administrative divisions that comprise each entity. For applications involving the interchange of geospatial information requiring the use of country codes, the following standard is mandated:

- *FIPS PUB 10-4, Countries, Dependencies, Areas of Special Sovereignty, and Their Principal Administrative Divisions*, April 1995 as modified by *Change Notice No. 1*, 1 December 1998; *Change Notice 2*, 1 March 1999; *Change Notice No. 3*, 1 May 1999; *Change Notice No. 4*, 25 February 2000; *Change Notice No. 5*, 10 August 2000; *Change Notice No. 6*, 28 January 2001, and *Change Notice No. 7*, 10 January 2002.

- Information transfer. This includes a mandate that the DoD use the open systems standards used for the Internet and the Defense Information System Network (DISN).

Mandated Information Transfer standards relevant to Web Services are:

- *IETF RFC 2616, Hypertext Transfer Protocol HTTP/1.1*, June 1999.
- *IETF RFC 1738, Uniform Resource Locators (URL)*, 20 December 1994.
- *IETF RFC 2396, Uniform Resource Identifiers (URI), Generic Syntax*, August 1998.
- *MIL-STD-2045-47001C, Connectionless Data Transfer Application Layer Standard*, 22 March 2002.

Emerging Information Transfer standards relevant to Web Services include Internet Protocol Version 6 (IPv6) and Mobile Host Protocol (MHP). IPv6 will provide better internetworking than the current IPv4 protocol, and will include support for the following: expanded addressing and routing capabilities, authentication and privacy, auto-configuration, and increased quality of service (QoS) capabilities. IPv6 is described by the following standards:

  - *IETF RFC 2373, Internet Protocol, Version 6 (IPv6) Addressing Architecture*, July 1998.
  - *IETF RFC 2374, Internet Protocol, Version 6 (IPv6) Aggregatable Global Unicast Address Format*, July 1998.
  - *IETF RFC 2460, Internet Protocol, Version 6 (IPv6) Specification*, December 1998.
  - *IETF RFC 2461, Neighbor Discovery for IP Version 6, (IPv6)*, December 1998.
  - *IETF RFC 2462, IPv6 Stateless Address Autoconfiguration*, December 1998.
  - *IETF RFC 2463, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*, December 1998.

  Mobile Host Protocol allows the transparent routing of IP datagrams to mobile nodes in the Internet. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. The following are MHP standards:

  - *IETF RFC 2507, IP Header Compression*, February 1999.
  - *IETF RFC 2794, Mobile IP Network Access Identification Extension for IPv4*, March 2000.
  - *IETF RFC 3344, IP Mobility Support for IPv4*, August 2002.

- Information modeling, metadata, and information exchange. This consists of activity, data, and object modeling, and information standards, including message formats. This encompasses the DoD Command and Control Core Data Model (C2CDM) and the Defense Data Dictionary System (DDDS).
  - Extensible Markup Language (XML) based information is the generally accepted choice of industry data/metadata interchange and is vital to the DoD interoperability strategy. XML is widely used for metadata definition, management, and exchanges. Integrating XML with middleware technologies and core database technologies provides the capability to exchange DoD mission-area data among heterogeneous environments.
  - In order to facilitate interoperability, the DoD has established the DoD XML Registry (http://diides.ncr.disa.mil/xmlreg/user/index.cfm) for collection, storage and dissemination of XML components. The DoD XML Registry is designated to be the single authoritative DoD repository for these components. System developers using XML for public interface are required to consult XML Registry before creating new components and reuse existing XML where practical.
- Human-computer interface. This is a common framework for Human-Computer Interface (HCI) design and implementation in DoD systems, the objective of

which is the standardization of user interface implementation options, allowing applications to appear and behave in a consistent manner.

- Information security. This consists of the standards and protocols to be used to satisfy security requirements.
    - The Intrusion Detection Message Exchange Format (IDMEF) is intended to be a standard data format that automated intrusion detection systems can use to report alerts about events that they deem suspicious. This format is described in *Data Model and Extensible Markup Language (XML) Document Type Definition*, 18 September 2001.

In addition to the JTA Core, the JTA defines special standards as required for each of the following domains:

- Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR). This domain includes activities where the primary focus is on one or more of the following functions:
    - Support properly designated commanders in the exercise of authority and direction over assigned and attached forces across the range of military operations.
    - Collect, process, integrate, analyze, evaluate, or interpret available information concerning foreign countries or areas.
    - Systematically observe aerospace, surface or subsurface areas, places, persons, or things by visual, aural, electronic, photographic, or other means.
    - Obtain, by visual observation or other detection methods, information about the activities and resources of an enemy or potential enemy, or secure data concerning the meteorological, hydrographic, or geographic characteristics of a particular area.
- Combat Support (CS). This domain addresses those specific elements necessary for the production, use, or exchange of information within and among systems supporting personnel, logistics, and other functions required to maintain operations or combat. This domain consists of automated systems that perform combat service support and administrative business functions, such as acquisition, finance, human resources management, legal, logistics, transportation, and medical functions.
- Modeling and Simulation (M&S). This domain provides a set of standards affecting the definition, design, development, execution, and testing of models and simulations. Modeling and simulation ranges from high-fidelity engineering simulations to highly aggregated, campaign-level simulations involving joint forces.
- Weapon Systems (WS). This domain covers those systems that are defined as a combination of one or more weapons with all related equipment, materials, services, personnel, and means of delivery and deployment (if applicable) required for self-sufficiency. Weapon systems have special attributes (e.g., timeliness, embedded nature, space and weight limitations), adverse environmental conditions,

and critical requirements (e.g., survivability, low power/weight, and dependable hard real-time processing) that drive system architectures and make system hardware and software highly interdependent and interrelated.

4 Relevant C2 Programs

4.1 Horizontal Fusion

The Horizontal Fusion (HF) Portfolio Initiative was created early in 2003 to respond to Secretary of Defense Donald H. Rumsfeld's vision of Force Transformation. The term "Horizontal" refers to reach across traditionally stove-piped organizations, and "Fusion" refers to the process and applications that allow Net-Centric integration and aggregation of information. Through Net-Centricity, users are able to seek the information they need across the battlespace through smart pull and through information sharing channels. The ultimate goal of HF is to allow discovery of and access to the right information at the right time by the right people regardless of mission.

HF Net-Centricity is made possible by new technologies and capabilities:

- The Bandwidth Expansion (BE) program, or Global Information Grid (GIG) BE, which provides a secure, robust, optical IP terrestrial network
- Joint Tactical Radio System (JTRS), which offers a family of software reprogrammable radios based on an open-communication architecture that will provide interoperable tactical wideband IP communications capabilities
- Wide-band SATCOM, which provides ubiquitous communications with optical quality bandwidth to mobile and tactical users
- Net-Centric Enterprise Services (NCES), which supplies the infrastructure and services to support the broad range of applications and data used in a Net-Centric enterprise
- Information Assurance, which is vital to support all efforts to ensure that the network is robust, reliable, and trusted
- HF, which provides Net-Centric applications and content needed to assure analysts and warfighters with the ability to make sense of complex and ambiguous situations across the battlespace.

The HF Portfolio Initiative was launched by DoD's Office of ASD/NII - CIO to accelerate the transition of Net-Centric Warfighting from the GIG Architecture vision to reality. Leadership for the Initiative realized from the outset that certain tenets would be essential to success:

- Only handle information once – Entering data multiple times increases cost and creates inefficiencies as well as data re-entry errors in both combat and business operations. The process of information sharing needs to be re-engineered in such a way that it is posted once and used many times in its original form.
- Post before processing – Access to data for cross-functional use is not delayed by unnecessary processing. Information providers have the responsibility to post information before using or manipulating it. Consumers can securely access the information they are cleared for as soon as it is collected.
- Smart pull – The environment shifts from system-centric (pushing data point to point regardless of quantity or need) to user-centric (pulling the relevant data to

solve problems). The consumers of information are smarter than their sources about what they operationally need, and they should be able to locate and pull it to them.

- Collaboration – Subject matter experts from diverse units or organizations are frequently called upon to come together to make sense out of special situations.
- Trusted network – Diverse, large-bandwidth pathways with security designed into the network and systems from the beginning are a must. Information assurance and interoperability must be the rule, not the exception.

The HF Portfolio process invests in initiatives that are DoD programs of record, as well as promising emerging technologies, which can be accelerated to Net-Centric operation. The Portfolio web-enables the initiative and weaves these initiatives into an information tapestry called the "Collateral Space." Investment in the HF Portfolio focuses on dispersing risk and increasing return on investment through the diversity of the Portfolio initiatives in terms of risk, maturity, and value to Net-Centric operations.

In FY 03, HF has accomplished integration of multiple technologies that provide Warfighters with real-time collaboration, sense-making tools, and intelligence capabilities at the Secret Level. Through the web-enabled portal called MARS, the Portfolio has integrated and demonstrated viable capabilities that will transition directly into the operations theaters.

In FY 04, the Portfolio will expand the capabilities established in FY 03 with a particular focus on:

- Cross-domain information sharing;
- Inclusive technical standards that allow for participation in the Portal environment by an increasing diversity of technologies;
- Extension to handheld wireless capabilities;
- Information assurance through PKI certifications and meta-data standards; and
- Enabling next-generation of the current FY 03 Portfolio technologies.

4.1.1 Web Services in HF

The MARS Portal uses existing market-driven, standards-based information technology, such as Web Services, portlets, Universal Description and Discovery Interface (UDDI), and metadata to ensure display, access, and data interoperability. Where no solutions currently exist, HF is developing open architecture standards that could, in turn, drive the future market and DoD technology base. HF FY-03 examples include the Federated Search capability and a systems-neutral track data model.

4.1.2 Value to C2 Users

Through the HF MARS Portal, forces can access and use new and existing data sources, services, and tools. Achieving "Power to the Edge" requires not merely information superiority but decision superiority. MARS provides the Net-Centric foundation for decision superiority by making information available on a network that people depend on and trust and populating the network with new, dynamic sources of information to defeat the enemy while denying the enemy advantages and exploiting their weaknesses. The first implementation of HF provides the framework and service-oriented architecture supporting net-centric development and operations. The core services include the following.

- Enterprise management services – provides end-to-end GIG performance monitoring, configuration management and problem detection/resolution.
- Messaging services – provides the ability to exchange information among users or applications on the enterprise, such as email, DoD-unique message formats, message-oriented middleware, instant messaging and alerts.
- Discovery services – provides processes for discovery of information content or services that exploit metadata descriptions of IT resources stored in directories, registries and catalogs.
- Mediation services – helps broker, translate, aggregate, fuse or integrate data.
- Collaboration services – allows users to work together and jointly use selected capabilities on the network.
- Application services – provides infrastructure to host and organize distributed online processing capabilities.
- Storage services – provides physical and virtual places to host data on the network with varying degrees of persistence.
- Security services – provides capabilities that address vulnerabilities in networks, infrastructure services or systems.

User assistance services – provides automated helper capabilities that reduce the effort required to perform manpower intensive tasks.

4.2 GIG Bandwidth Expansion (GIG BE)

Global Information Grid Bandwidth Expansion (GIG-BE) is intended to create a ubiquitous "bandwidth-available" environment for enhanced communications of information for national security intelligence, surveillance and reconnaissance, and command and control information-sharing. The GIG- BE will provide approximately 100 times the current telecommunications capacity to critical Defense sites around the world.

GIG-BE is part of a larger DoD initiative known as “Transformational Communications.” Transformational Communications is composed of three fully-integrated segments:

- The terrestrial segment will be based upon fiber optics and include the GIG Bandwidth Expansion.
- The wireless or radio segment will be based upon the software programmable Joint Tactical Radio System.
- The space-based segment will be composed of several systems with the Advanced Wideband System serving as a gap-filler while we pursue the objective Transformational Communications Satellite capability.

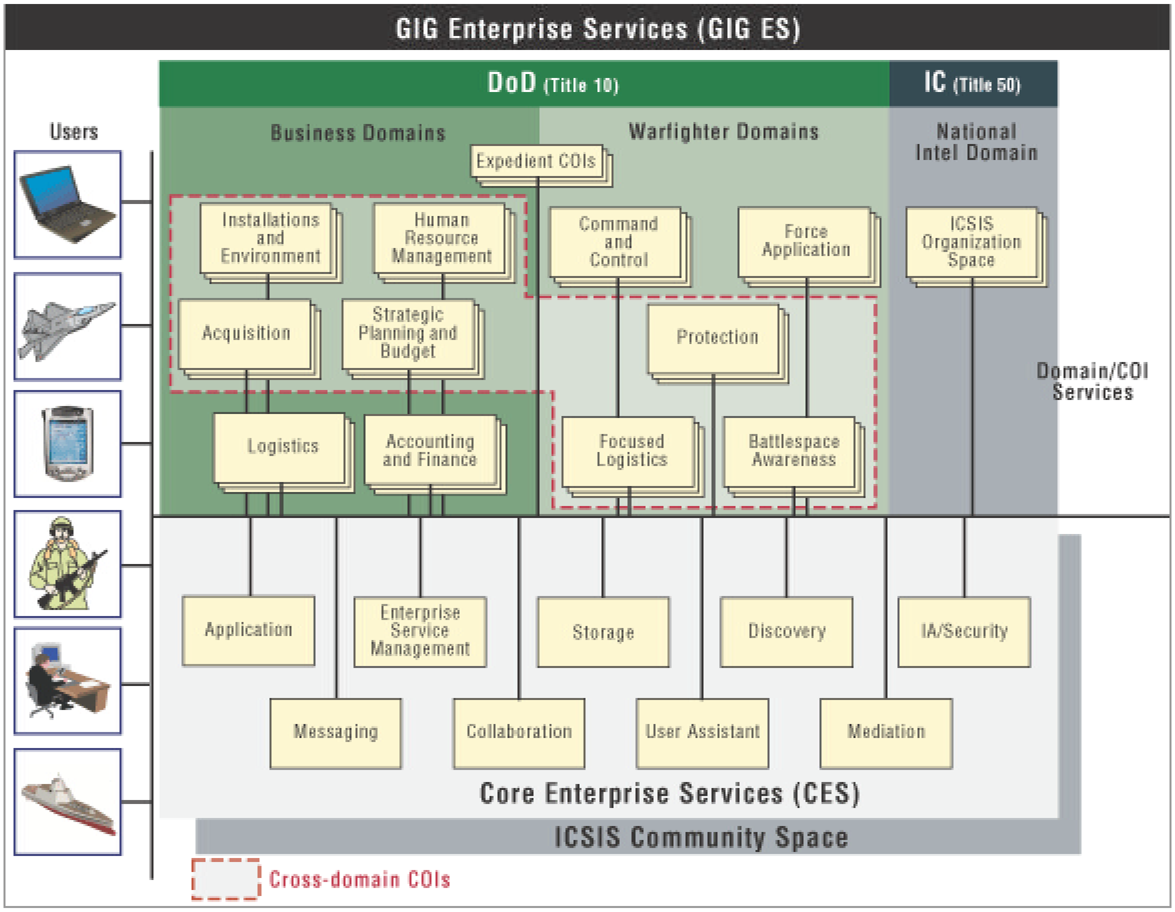
The Defense Information Systems Agency (DISA) is implementing GIG-BE by aggressively enhancing their current end-to-end information transport system, the Defense Information System Network (DISN), significantly expanding bandwidth and physical diversity to selected locations worldwide. The vast majority of these locations will be connected by a state-of-the-art optical mesh network design.

The GIG-BE program provides the network hardware and communications pathways to support DoD surveillance assets, reach-back, sensor-to-shooter integration, collaboration, and enterprise computing. Increase bandwidth will eventually support new applications of net-centric software for improvements to self-synchronization, shared situational awareness, sustainability, speed of command and action, and full access to a rich and enabling set of information assets from the battlefield.

The GIG-BE does not directly address the implementations of web services, though some network systems (e.g., intelligent routers) may include routing, messaging, and security optimizations enhanced by the foreknowledge of today's common practices, including web services technologies and service oriented architectures. Web services developers and net-centric system architects should take GIG-BE into account when designing systems and architectures, not to take bandwidth for granted, but to wisely leverage the resources that are and will be available. However, it should also be pointed out that some systems and applications will still be bandwidth-constrained for the foreseeable future, especially in highly-mobile environments (including ground, air, and sea operations).

4.3 Network Centric Enterprise Services (NCES)

Network Centric Enterprise Services (NCES) is a program for providing enterprise services in support of the Global Information Grid (GIG). These GIG enterprise services (GES) are divided into two categories, community of interest (CoI) services, and core enterprise services (CES). The following figure shows this division.



Source: http://ges.dod.mil/about/solution.htm

Many of the CES are provided by existing systems operating throughout DoD. Others will be based on state of the art COTS products, whenever possible. Under GIG ES, these will be available to all components, deployed by users across DoD in a consistent manner. This will enable leveraging of best-of-breed concepts (many of them Web-based) and will maximize the net-centric performance of the GIG.

The CES will support 4 different styles of service interaction. These are:

- Request/Response - This is a single interaction usually initiated by the consumer of the service. The consumer creates a logical request for the service and the service answers back with a response. An example might be a lookup or validation service, such as the Federal Express service to lookup a package's shipping status.
- Stream - In this model a continuous stream of information is created by the service provider. The consumer connects to the stream as needed. Commercial examples include video servers, and financial market indicators.
- Publish/Subscribe - Here a consumer registers with the service provider to receive events on a logical device, often referred to as a "channel". Events are published onto the channel, and subscribers receive the events. The events can be complete data objects with fields and attributes. Crossing multiple channels with business rules can allow for high-value events to be generated. For example, events from a

"one-way plane ticket" channel could be combined with a "wanted list" channel to produce events of significance.

- Threads/Process - A series of sequenced interactions between a service provider and a consumer. Often a set of defined structured messages is used to move the provider and consumer through a business process.

The Request/Response interaction model will support Web services.

4.3.1 Applying Web Services

Some of the CES's may be implemented as Web services, in some configurations. On the other hand, many of the CES's will provide support for Web services that provide COI services.

The following sections discuss each of the core enterprise services from the point of view of Web services user requirements.

4.3.1.1 ESM NetOps

The Enterprise Service Management NetOps Service (ESM-NetOps) enables the life cycle management of the information environment and supports the performance of the NetOps activities necessary to operationally manage information flows in the information environment. A key underlying tenet of ESM/NetOps is that all CES and CoI services must be "manageable" in all deployed operational environments. This means that they must be equipped or instrumented with the appropriate set of built-in functional management capabilities and that they must support agreed upon operational policies, processes and procedures. This will have an impact on every Web service that is part of the GES.

In order for a Web service to be manageable, certain supporting functionality has to be built into the service, at development time. Currently there is an OASIS Web services standard effort (OASIS Web Services Distributed Management TC, WSDM, at http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm) that – among other things – is trying to define the functionality that will have to be built in to any manageable Web service. At least two of the organizations participating in this effort have declared that they have patent rights that may be infringed by compliant implementations, depending on the specification that emerges.

Additionally, ESM/NetOps itself will have the requirement that Web services implement certain functionality, in order to be compliant. Web services will have to be able or report at least:

- The activity of critical processes and resource utilization and accurately and securely report anomalous behavior that breaches agreed upon thresholds
- Their operational configuration and accurately and securely report any changes in configuration or operational status
- Their overall operational performance and accurately and securely report any failure to meet agreed upon service level agreements

- Their security status and to accurately and securely report on any changes in security status to include any anomalous security behavior that could be indicative of a cyber-attack directed against the service

Implicit in these requirements is the notion of a resource state model. WSDM may define a resource state model that is compatible with the one that ESM expects, or (as of this writing) they may not define one at all.

Some Web services will run inside an Application Server (see, below) and this environment may handle some of the required ESM support functionality.

The Web Services Standards Analysis Report covered an OASIS standards effort relevant to ESM/NetOps. To summarize:

- WSDM – chartered to develop the model of a web service as a manageable resource.

This standards effort is scheduled to come out in draft form in January 2004. We recommend that DISA review the progress of this standard, because it is addressing the issues that ESM/NetOps needs to address. Nevertheless, there may be mismatches between the ESM/NetOps requirements and the WSDM requirements. It is recommended that DISA closely monitor the overlap between ESM requirements and the WSDM activity.

4.3.1.2 Messaging

Messaging will provide services to support synchronous and asynchronous information exchange. Aside from standard e-mail protocols, there are currently no relevant Web services standards in this area. For instant messaging, currently none of the efforts to standardize this area have been able to reach consensus. This may be because some large commercial entities feel that standardization of instant messaging would be counter to their commercial interests.

4.3.1.3 Discovery

Discovery will provide the set of services that enable the formulation and execution of search activities to locate data assets (e.g., files, databases, services, directories, web pages, streams) by exploiting metadata descriptions stored in and or generated by IT repositories (e.g., directories, registries, catalogs, repositories, other shared storage).

Because it currently has a stable version 2, UDDI can provide some of the functionality that is needed for discovery. UDDI version 4, which is currently underway, is looking into providing more support for the semantics of service discovery. The metadata model of UDDI, which provides the basis for searching a UDDI registry, is somewhat different from the metadata model for NCES. Because of this, the current UDDI is less useful than it could be.

4.3.1.4 Mediation

Mediation will be the set of services that will enable transformation processing (translation, aggregation, integration), situational awareness support (correlation and fusion), negotiation (brokering, trading, and auctioning services) and publishing.

Web services standards for business-to-business interaction, including the emerging W3C standard for choreography, will be relevant to the Mediation ES.

4.3.1.5 Collaboration

Collaboration will allows user to work together and jointly use selected capabilities on the network (i.e., chat, online meetings, work group software etc.) Collaboration in the sense of electronic business exchanges is part of the Mediation ES. The Collaboration CES deals with collaboration in the sense of group coordination.

Current commercial collaboration products are either thick client or thin client based. The thick client solutions require the user to download and install client software. The thin client solutions are browser base, and typically use either applets or other browser extensions. In either case, Web services have not played an important part in this area.

4.3.1.6 User Assistant

The User Assistant CES will provide automated capabilities that learn and apply user preferences and patterns to assist users to efficiently and effectively utilize GIG resources in the performance of tasks.

Goal-oriented software can assist users in discovering GIG services that will be useful, and in connecting those services together. Currently such software is based on planning technologies, and requires semantic knowledge. The DAML-S (OWL-S) group is defining a standard language for representing services in a way that planning software can utilize the services. Additionally, the W3C OWL language provides a standard syntax for providing the semantic information that the User Assistant CES will need.

4.3.1.7 Security

The Security CES is the set of services that provide a layer of Defense in Depth to enable the protection, defense, integrity, and continuity of the information environment and the information it stores, processes, maintains, uses, shares, disseminates, disposes, displays, or transmits. As pointed out in the first part of this report, a number of Web services specifications deal with security. However, none of them yet deal with the MLS that the Security CES will need.

4.3.1.8 Storage

The Storage CES will provide the set of services necessary to provide on demand posting, storage and retrieval of data.

Storage of XML data in a way that it can be queried will be important, due to the large volumes of XML data that will be present in the GIG. The W3C XQuery language is set to become the standard for this kind of querying. (Currently there is no standard language for XML data manipulation operations, such as XML update.)

The DoD XML repository provides some storage for XML data, but there the focus is more on structural data (XML Schema) and on metadata.

4.3.1.9 Application

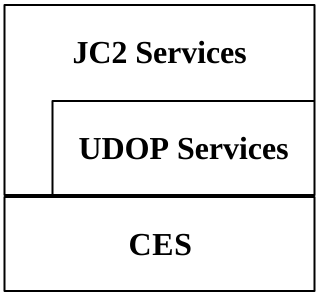
The set of services necessary to provision, host, operate and manage the GIG ES assured computing environment. Part of what the Application CES provides will be support for Web services. Depending on the configuration, this may be anything from a simple servlet engine to a full Application Server. In the first case this would require a compliant Web service to implement a lot of additional functionality, e.g., for management and for IA; in the second case the additionally functionality would be provided by the Application Server.

4.3.2 Transforming from the Common Operating Environment (COE)

By incorporating NCES-provided services into their systems, COE 4.7-based systems become NCES-based systems. One way to incorporate NCES services is by using Web services.

4.3.3 User-Defined Operational Picture (UDOP)

As part of the net centric transformation of the COE, a transformation that will move control out to the users at the edges of the C2 topology, the Common Operational Picture (COP) will transform into the User-Defined Operational Picture (UDOP). UDOP will be a composable set of network services that depend on the CES services, and that in turn support the higher-level JC2 services that will display an operational picture. A typical function of the UDOP services will be to fuse data.



Dependencies between GES Service Layers

The UDOP services can be exposed in different ways, and one of these ways is as Web services. The XTCF case study below discusses one way in which this can be done. XTCF and UDOP development efforts are now being integrated with equal cooperation between Navy and DISA project teams and their leadership.

4.4 Joint Command and Control (JC2)

Web services are well suited for service oriented architectures by the very nature of their construction, and the desire for service-oriented architectures is arguably one of motivations behind the rapid development and adoption of web services and related technology. The JC2 strategy clearly states service oriented architecture is a key enabling factor for achieving the goals of the effort. The primary drivers for service-oriented architecture in JC2 are the high degree of development flexibility it provides, the minimization of software deployment dependencies and independence from specific hardware and software platforms. Creating cohesive systems from these disparate parts presents unique challenges to the developers and organizations charged with the task. Many of the capabilities taken for granted in simpler stovepipe systems require sophisticated new implementations and adherence to carefully thought out standards in order to achieve the needed interoperability. Fortunately these challenges are not unique to the JC2 strategy. The industry has already invested tremendous effort and resources into finding ways to enable web services to meet these ends.

4.4.1 Applying Web Services

The supporting tools and specifications required to make web services a viable platform for JC2 implementation are either available now, or moving rapidly into advanced states of specification and adoption. Beyond simply providing a service-oriented architecture, several other key capabilities must be present to make web services a viable means for achieving the goals of JC2. These include platform and vendor neutrality, the ability to provide security for services and the messages they exchange, rigorous description of service interfaces and messages, and capabilities for service description and discovery. Standards that are currently emerging in the field of web services promise to provide the core capabilities required for the more advanced needs of the JC2 strategy. These needs include transactional capabilities, workflow capabilities and advanced visualization capabilities.

The importance of platform and vendor neutrality for the development of JC2 cannot be overemphasized. Many of the application requirements of JC2 are already implemented in legacy systems, particularly GCCS. GCCS components are built using a wide variety of software tools and products. The difficulties of exposing these capabilities in a service oriented architecture will be significantly simplified by providing the responsible developers greater freedom to choose appropriate software tools and products in order to achieve that goal. Developers will be able to choose tools to expose GCCS capabilities based primarily on the ability of those tools to interoperate with the components they are exposing rather than their ability to interoperate with a proprietary communication standard.

Web services also provide easily testable components. Because message formats and service interfaces are well defined and services are exposed on the network they can be easily tested by independent tools. The fact that these services are easily tested reduces the risk of defects being unintentionally introduced, prevents the need for large amounts of testing code to be imbedded in the service implementations, and makes it easier to assert that services are working correctly after major modifications or being completely replaced. These characteristics are considered critical to employing agile development methods intended to provide superior responsiveness to user needs, shorter development cycles and higher quality software components.

In addition to these general characteristics there are many particular specifications related to web services that enable the development of critical elements of the JC2 strategy. The remainder of this section discusses many of these specifics by explaining how particular web service related specifications and concepts address several specific needs of JC2. The specifications being discussed are at various levels of acceptance and standardization. It is important to leverage standardized technologies where possible so that commercial components can be used to reduce cost without being locked into any particular vendor's solution. This ability to change implementations allows features for scalability, reliability and availability to be swapped in and out as new needs arise and previous needs evolve.

4.4.2 Creating Mission Capability Packages from Components

Registry capabilities provided by UDDI and ebXML provide the fundamental capabilities required for rapid development of Mission Capability Packages and enabling users to define their own operational pictures for their individualized tasks. These specifications differ somewhat, but the fundamental capabilities are essentially the same. A closer look at UDDI will highlight the capabilities that make these modern registry services suitable for systems that require dynamic discovery and high-availability.

The API provided for searching UDDI provides mechanisms for wildcard based search to gather many results as well as detailed ways to specify precise API versions desired for a given service. We will focus on these two capabilities in order to demonstrate how they can be used to help facilitate a User Defined Operational Picture. The various "find_X", API calls provided by UDDI supports a variety of search criteria with flexible semantics that allow sophisticated queries to conduct searches of the repository in a single call. This capability allows end users (through the use of specialized search applications) to search for services related to specific categories of functionality, published by specific entities or following particular naming conventions. In addition to this capability, and of particular importance in situations where services will be dynamically aggregated, is the ability to specify zero or more tModel parameters to the calls "find_business," "find_service," and "find_binding." TModels are used to specify the technical requirements a service must meet in order to be considered a match for a given query. This is also known as service's technical fingerprint, or simply fingerprint. By the specification of tModel parameters queries may be constrained to particular technical subsets. If the specification for a service is separated into several sections, and each one assigned its own particular fingerprint, then searches can be constrained to complete or partial implementations of the specification. Fingerprints are also used to specify information about particular versions of specifications, thus addressing concerns about clients attempting to bind to incompatible versions of components.

When this component discovery is automated it also allows for fault tolerance. If the service a client is currently using fails in some way it is then possible to search for a new component that meets the minimum required technical capabilities. Obtaining fault tolerance via this kind of redundancy allows individual components to be more easily taken down for maintenance, and for new implementations to incrementally replace older ones.

Although the UDDI and ebXML specifications are similar, more implementations are available for UDDI and it is also more widely used (at least in the United States). Unless some particular benefit is derived from using the additional capabilities of the ebXML registry then UDDI should be used. Being a core capability special attention should be paid to the performance, scalability and availability features of a UDDI server, and any UDDI services should be deployed redundantly. The behavior of these components will have direct impact on the performance of JC2 as a whole.

4.4.3 Creating Mission Capability Packages from Components

Mission Capability Packages (MCPs) are aggregations of service-oriented components collected to assist a user in accomplishing a specific task. The components assembled to create these packages will likely change over time, and the precise nature of tasks will evolve with the needs of the communities of interest. As such it is critical that Mission Capability Packages can be quickly constructed and modified based on the immediate needs of a given task and the users assigned to it. The capabilities of modern service registries to make the individual components available and searchable are detailed above. However, these capabilities alone are not adequate to meet the goals of a MCP.

MCPs must also understand the dynamic interfaces of services and be able to describe workflow between these dynamic interfaces to create meaningful interaction among the components being aggregated. The static interface of a web service (e.g. as described by WSDL) is not adequate to completely describe the service's expectations when involved in an interaction with some client. This dynamic interface describes valid sequences in which methods can be invoked as well as other interdependencies between a service's operations. With these details described a service may be treated in a more abstract manner, making it easier to understand how a particular service would participate in a more complicated workflow. WSCI (Web Service Choreography Interface) provides the capability to unambiguously describe the dynamic interface of a component including (but not limited to) characteristics involving exception handling, transaction boundaries and alternate behaviors based on the runtime values of messages (it is important to note that WSCI does not describe the way in which transactions are conducted, but merely the boundaries of transactional behaviors). When WSCI is used to collaborate with a workflow tool it allows the precise description of interactions and boundaries of each participating service.

On top of this dynamic interface layer it is then possible to build a workflow that integrates many components together collaborating to perform a complex multiple step task. A key difference between the description of workflow and dynamic interface is that workflow does not describe all possible interactions, but a set of paths that are followed to meet particular desired outcomes or rollback in failure scenarios. This is in contrast to the description of a dynamic interface, which is expected to be an exhaustive coverage. In this way the workflow layer's descriptive needs overlap with that of executable programming languages. An example of this currently undergoing specification is the Web Service Business Process Execution Language (BPEL4WS). This XML language contains many constructs familiar to programmers. Included in this are conditional constructs, looping constructs, invocation of external services, and creation of scoped blocks that hide data from external scopes. Unfortunately BPEL4WS is not aware of WSCI, and interactions between layers of this type, although being aggressively pursued, are still in their very early phases.

Well-established standards groups have not yet finalized key elements of these specifications. Special attention should be paid to the development of standards for these areas in the near future in order to guide decisions on what developing standards to choose. Mission Capability Packages are a key element of JC2 and decisions about these standards will significantly influence the architecture of MCP implementation. With MCPs performing such a critical role in the JC2 architecture this does create some difficulty, and

MCP development is likely to move ahead before final decisions are made about competing standards in this area. Where possible, especially in early development, the design of MCP components should remain flexible enough to change what specifications are used for dynamic interface description and workflow.

4.4.4 Web Service Security

The primary focus in this section will be to cover details concerning the application of standardized technologies to security issues of authentication, authorization, identity management, integrity, confidentiality, non-repudiation, trust and policy as they may be applied to JC2. Using these fundamental elements it is possible to define rich security protocols for secure web services, protect resources, secure content traveling on the network and provide single sign on capabilities so that security contexts can be shared by many services during an interaction. These fundamental capabilities are addressed by a number of emerging standards including WS-Security, XACML and SAML.

JC2 requires support for levels of access as well as constraining access to various compartments based on a need to know or a need to hide sensitive information. The general capabilities associated with Public Key Infrastructure (PKI), as leveraged by web service security standards and proposals, provide an extremely powerful infrastructure to address issues of identity, integrity, non-repudiation and confidentiality. Through the use of asymmetric cryptography (a.k.a public key cryptography) it is possible to positively identify the source, verify the integrity, and insure the confidentiality of a message. The use of this form of cryptography is based on the simple idea of two separate keys, one public and one private, that can each reverse encryption operations conducted using the other, where knowledge of the public key does not imply knowledge of the private key. With this latter condition being the case, the public key can be freely distributed. One key additional element of a complete PKI is that of a Certificate Authority (CA). The role of the CA in transactions using PKI is to act as a trusted third party that can vouch for the identity of a person or entity associated with a public key. The level of trust associated with any public key is limited to the level of trust given to a particular CA. It is important to consider the integration of many certificate authorities with varying levels of trust associated with them. Clients of secured services may be denied access simply because they are not associated with a certificate authority with an adequate level of trust.

Using these capabilities it is possible to securely transmit messages or portions of messages, verify the sender of a message or response, and ensure that messages were not tampered with in transit. When combined with traditional access control constructs like Access Control Lists (ACL) the fundamental capabilities needed to secure individual resources are fully represented. These are the same capabilities leveraged by the specifications listed above to accomplish their various security goals. Systems as well as users will be required to have their own asymmetric keys that can be used to verify their identity. See also *Section 3.1, Security* for related information.

The number of implementations in this area is growing rapidly. SAML in particular has been widely adopted by industry and many implementations are available. The same can be expected of many of the other specifications in the near and medium term. In this area it is important not to think of the specifications as exclusive. Many of the specifications that address security can be used in complimentary ways. When the security needs of different services require features offered by several specifications it will often be possible to aggregate these capabilities to achieve the needed security.

4.4.5 Transforming from the Global Command and Control System (GCCS)

As stated earlier much of the application functionality needed for JC2 already exists in legacy systems. In its current form, updates come infrequently, and as part of time consuming and expensive upgrades. There are a variety of reasons for this, not the least of which are COE compliance and tight coupling of components. That being said it is still desirable to reuse components of GCCS where possible in order to more rapidly implement JC2 services. Utilizing these well-tested components will save a significant amount of time and money. When components utilizing GCCS components are properly hidden behind web services they can later be replaced without impacting Mission Capability Packages that depend on them.

Many GCCS components may be wrapped with JC2 service infrastructure in order to provide services. A wide variety of tools are available for many different programming languages and operating systems to accomplish this goal. The service-oriented nature of JC2 makes the details of operating system and implementation language irrelevant allowing more GCCS components to be ported more easily than would otherwise be possible. Even with these tools some GCCS components will not be suitable for direct migration. This same benefits and pitfalls apply to the addition of new services to JC2 as well. This is one potential route to getting greater value from third party components within JC2.

Many of the services that were once provided in LAN environments as parts of GCCS will be moved to more distributed environments and become more centralized, with a much larger number of clients utilizing instances of a given service. For example, thousands, or tens of thousands, of clients (including other services) may utilize an implementation of TMS that has been migrated to JC2. There are many strategies that are available to build services that are scaleable and highly available, but it can be difficult to implement them completely if they are not carefully planned for. Getting the full benefit of these strategies does place constraints on the designs that can be used. Each approach has its own strengths and weaknesses, and should be chosen based on the needs of a particular service.

One common strategy for creating scalable services is the use of clustering. The general idea is to spread the required processing horizontally across many pieces of commodity hardware in such a way that an increase in demand can be addressed by the addition of hardware. Each node is a complete version of the application, essentially an exact replica of every other node in the cluster. This approach also provides redundancy, which allows individual instances to be removed from the cluster (for maintenance or as a result of some malfunction) without seriously affecting the overall operation of cluster. This technique is widely supported in application servers, but is less common in database servers. If the database supporting a given service is not clustered along with the application servers it represents a potential performance bottleneck as well as a single point of failure. There are practical limits on the number of machines that can participate in a cluster where application state is maintained. Most services must persist data at some level, and at least some portion of that data is shared with other clients of the service (consider TMS for this example as well) and is subject to editing under the rules of ACID transactions.

This requires nodes in the cluster to be aware of one another in order to share state, and participate in two-phase commits during edits. When shared data is updated in one node other nodes must be notified as part of the ACID update which leads to an increase in the runtime of the operation. Application servers take a variety of approaches to optimize this behavior, including the user of multicast sockets and partitioning of large clusters into subsets to limit the scope of transactions and replication.

Another strategy for distributing the workload for a problem is to partition the problem space for a given service across several physical servers. In this case some attribute, or combination of attributes, is used to determine where requests for services related to a particular data item will be served. This approach also requires the technique to be applied to every tier of the service in order to allow the system to scale. Unlike the clustering approach, partitioning does not posses the additional benefit of redundancy. The failure of any given node in a simple partitioning scheme means that information being served by that node is no longer available. This problem can be remedied with a hybrid approach, where a cluster of machines represents each partition, but this does not provide benefits beyond those offered by the simple clustering approach detailed above unless some practical upper limit has been reached on the number of machines operating in a cluster. As detailed above, clustering software generally has its own means to deal with clusters containing many nodes.

Another concern that should not be ignored in the transition from LAN based services to WAN based services is performance. Two critical characteristics influence performance, bandwidth and latency. Because JC2 is being built on next generation network infrastructure problems of bandwidth are largely addressed, at least for the best-connected facilities. Many facilities will still have significant bandwidth limitations for applications that need frequent updates or otherwise frequently access services located on the WAN. There are practical limits on how far latency can be reduced. These limitations are based on the speed of transmission of data through wire, fiber or radio waves. Terrestrial distances can have significant impact on latency, and the effect is much more pronounced when geosynchronous satellites are part of a network route, creating latencies in the hundreds of milliseconds. In many situations these latency issues can be addressed by utilizing content staging. In this scenario the data most frequently used by an application is replicated closer to where it is actually needed to reduce latency. This technique can dramatically increase the performance for applications where the majority of the work consists of reading data. When updates of the persistent data are also required the use of this technique implies a need to pay special attention to the consistency of data. It is possible for a particular record in the master data store to be updated after the local cache of that same record data is updated. If the remote client then attempts to update that record based on local data that is stale, the master data store may (depending on application requirements) need to reject this update attempt. This type of data consistency strategy is often referred to as optimistic locking.

The techniques discussed in this section are not unique to JC2, and are widely used and supported in industry. No single solution is ideal for all scenarios and solutions should be selected only after careful consideration of application requirements. It is also important to test the functionality of these tools early and often as part of the development process. This prevents the late discovery of technical choices that could be in conflict with the constraints created by the choice scalability/high-availability/performance tools being used.

Transformation of existing GCCS components holds potential to expedite the creation of JC2, but it is not without its difficulties. Considerations of scalability and availability of services should be addressed as early in the process as possible, and possible solutions should not be adopted without prototyping and testing of these solutions. Many tools may be used to assist in wrapping existing GCCS services in JCS compliant services, but not all of them will be equal in their abilities to provide scalability and availability. As discoveries related to individual tools and software packages are made it is critical that these discoveries by made generally available among the entities tasked with migrating GCCS services to JC2.

4.4.6 Application of Emerging Standards to Long Term Goals

Several standards in the early stages of adoption promise to provide functionality that will be critical to a mature JC2 system. Of particular interest are transaction, coordination and context management capabilities. There are a number of emerging standards in this area, none of which are well established in the market at the time of this writing, but many of which are progressing rapidly. The capabilities represented by these various efforts will be key to enabling reliable and flexible collaboration in a service-oriented architecture.

Specifications that provide for a shared context (e.g., Web Service Context or WS-CTX) allow interactions between several different participants to share common information and common resources to work towards a common outcome. This capability is key to the JC2 environment as it is expected to function by aggregating loosely coupled components into cohesive MCP units for end users. Mechanisms for sharing of information and resources is required to allow these loosely coupled services to act cohesively. This notion of context allows the participants of an activity to scope work within this activity by utilizing a context object. Other critical services that will further enable JC2 will likely rely on a context service to manage their own activities. Examples of other core services that could be expected to utilize a context service are transaction, coordination and workflow services.

The notion of coordination between services provides a layer between a simple context service and more complex services such as replication, transactions, workflow and caching. Enabling such services requires a mechanism where all participants in some activity are notified of particular events of interest. In order for such a service to be widely useful it must allow generalized treatment of the messages it is to deliver so that it does not limit the usefulness of those messages to the recipients. By providing such a general coordination capability a coordination service would not only enable the above general needs, but could also be leveraged to coordinate arbitrary activities between the services that make up the JC2 infrastructure. By reusing a robust coordination component the services that make up JC2 can focus more clearly on their own function and delegate coordination activities.

Another aspect of JC2 currently being addressed by emerging standards is transactional capabilities. In order to insure the consistency of data in a service oriented architecture several components must be able to participate in the same transaction. Do to the distributed nature of JC2 and web services in general the situation is more complex than traditional atomic transactions as conducted by a relational database system. This requires a much more flexible protocol (or set of protocols) be available to support transactions. The normal two-phase transaction is still relevant in many scenarios, but long running cases (in minutes, hours or days) are reduced to unacceptably low levels of concurrency by the locking of resources in this case. These long running cases may not posses the same guaranteed ACID properties as their two-phase counterparts. In order to achieve the desired "all or nothing" outcome programmers must provide application specific actions for forward or backward error recovery. This is significantly more complex than typical ACID transactions, but it a side effect of the distributed nature of JC2. With these capabilities in place the long running actions can be considered atomic. This allows them to be combined with other long running actions and shorter ACID actions to form more complex transactions and wider reliable collaboration. It is reasonable to expect the context and coordination capabilities previously discussed to be used as building blocks to support this kind of transaction service. They could certainly be provided by the transaction service itself, but those services are also very useful in a variety of other roles making it desirable to offer them as stand-alone services.

It is likely new needs will be discovered as the JC2 architecture matures and implementation moves ahead. As this occurs the likelihood that similar needs have been discovered and addressed in standards communities should be considered, and standard solutions should be adopted for these problems when possible. This section is not intended to be an exhaustive coverage of all potentially relevant emerging web services standards, but rather to point out the utility of some current efforts in more advanced stages of specification. More thorough investigation of the detailed JC2 requirements and current and emerging web standards would certainly benefit the JC2 effort.

4.5 Rapid Prototype Insertion and Delivery System (RAPIDS)

Reusable Application Integration and Development Standards (RAPIDS) is a PEO C4I initiative that emphasizes the needs of sophisticated fleet users who need simple ways to assemble capabilities in order to meet rapidly changing mission requirements. The RAPIDS objective is to enable the extension of existing applications by mixing and matching components from various applications, building systems from a small number of existing components that are downloadable over the network or connecting to services that are hosted by a third party.

The objective will be met in two ways:

1) By providing a set of downloadable components, which will offer the ability to reuse technology across the PEO C4I/Fleet enterprise by providing components that can be easily connected in a wide variety of ways to provide new mission capabilities with minimal development effort and without requiring detailed knowledge of the internal workings and implementation details of an application.

2) By providing guidance to developers, which is not intended to replace or supercede the Task Force Web Developer's Guidance. In fact, it references that document for all developers building into the TFW portal environment. The intent of

the guidance is to provide specific architectural direction so that developers are able to construct their applications into services that can be easily adopted into a web portal; a basic, web native application; or Java environment with little or no re-coding.

The business case for RAPIDS (i.e. distributed network-accessible mission capability parts developed in an Open Source environment) is intuitively compelling. Implementing a controlled collaborative and distributed software development environment that exposes capability and source code to all developers in the enterprise will significantly increase software re-use, accelerate the integration of new technologies, and drive down time and costs required for developing and maintaining software, resulting in a dramatically increased "Speed to Capability."

4.5.1 Principles Guiding RAPIDS

The follow list outlines the principles guiding the RAPIDS initiative.

1. Standard, published interfaces
2. Separation of interface from implementation
3. Open architectures instead of closed architectures
4. Database independence
5. Joint interoperability
6. Uniformity in architecture and design
7. Recognizing and embracing diversity in the IT Enterprise[1]

RAPIDS will evolve over time, initially providing infrastructure and guidance, maturing into a fully populated RAPIDS library with an extensive collection of components that follow the guidance provided in the RAPIDS Developers Guide.

4.5.2 Web Services in RAPIDS

The RAPIDS initiative is embracing web services, devoting a good portion of the Developer Guide to web applications and web services. There are sections on SOAP, WSDL, UDDI, and Open GIS Consortium (OGC) specifications.

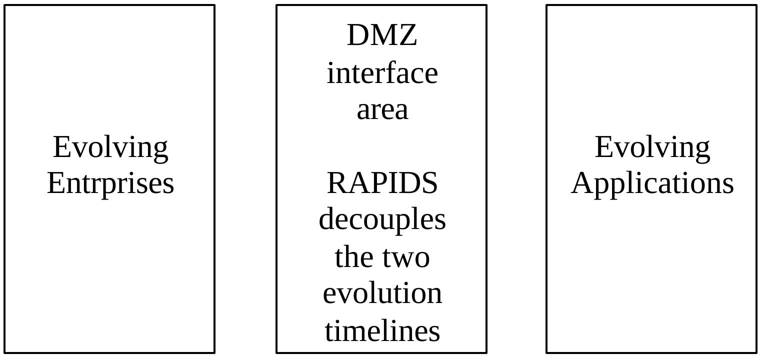
4.5.3 Value to C2 Users

RAPIDS will allow:

- The fleet to develop client side applications in the RAPIDS environment and "snap-in" to the Navy application infrastructure.
- Assembly of mission capabilities from mission capability parts through a business component approach.
- Maximum reuse of programs functionality in the Navy and Joint communities.

[1] **IT Enterprise:** An IT environment with a non-invasive distributed component based architecture in which applications exchange information in understandable formats and contexts.

- Architectural groundwork and migration strategies for existing applications to target multi-tier business component and services architecture.
- Implementation of connection strategies to extend life and reach of legacy applications while legacy application developers define future direction for their systems, reducing the restructuring burden.
- Mitigation strategies that decouple enterprise development from program application development. This creates a "DMZ" that permits independent paces of development and change on each side of the enterprise to reduce risk and impacts of changes to application developers.



This first phase of RAPIDS will focus on building capabilities using Open protocols, tools and standards. Future versions of the RAPIDS SDK will address how to hook into the PEO C4I enterprise to create more sophisticated client side applications as those portions of the enterprise become available and how to build capabilities using commonly available tools.

4.6 Task Force Web (TFW)

The Task Force Web (TFW) initiative began in 2001 with a goal to enable the Navy to The Task Force Web (TFW) initiative began in 2001 with a goal to enable the Navy to take advantage of the Navy Marine Corps Intranet infrastructure and eliminate redundant services.[2] Providing one single point of entry provided Command and Control users easy access to the variety of informational applications. Users could login to the Task Force Web portal and see applications. Users could then request accounts to the applications, which was maintained separately.

The Task Force Web initiative seeks to leverage industry provided solutions while remaining loosely coupled to a particular vendor's implementation. The current TFW portal system uses Computer Associates' Cleverpath portal and BEA WebLogic Server. Other elements of the TFW software load include:

- Windows 2000 Server
- Oblix Netpoint 6.1.1
- MS SQL Server 2000

[2] *McKenna, Ed. "Military Big on Portals", Federal Computer Week. 9 June 2003.*

- Computer Associates Cleverpath Portal 4.01
- BEA Weblogic 7.0
- IIS [3]

Future goals include integrating Single Sign-On (SSO) hosted by Oblix with the Navy network directory system. C2 users will experience a great benefit in ease-of-use through having one centralized account for all their network and application authentication needs. Users will also benefit in being able to login once rather than multiple times. The Navy gains the benefit in simplified system administration.

A majority of the applications hosted on TFW were developed as single stand-alone applications with little to no integration with other applications. The Navy is using the TFW initiative to facilitate a paradigm change from a client-centered/application-centered approach towards a services based approach. Most existing web applications in the TFW portal have their business logic tightly coupled with their presentation (the Graphical User Interface). In fact, in the case of some types of applications such as those executed in Domino Notes, ColdFusion, and Active Server Pages it is extremely challenging to separate them.

A Model View Controller (MVC) design pattern allows developers to separate presentation from data. The great advantage to an approach that accomplishes this level of separation is that it allows for greater integration across disparate data providers. This can have concrete real-world advantages for a C2 user. For example, one application may provide ship schedules, such as WebSked. Another application may provide ship engineering change kits. A third application may provide education to sailors. Rather than have a C2 user login to 3 different web applications to check on when a ship is in port, a webservice approach might allow integration of 3 separate data sources to the user. So the C2 user could find out when a ship was in port and schedule sailor training and a ship engineering change in a single web service.

TFW is embracing the services-centric approach by supporting the implementation of SOAP. It is also serving as a change broker in the Navy environment by encouraging developers to switch from a pure stovepiped application approach to exposing data as a service. TFW can then develop a SOAP application that can consume and present the various data sources. This vision is not currently implemented in the majority of existing applications in TFW's portal.

TFW is a crucial part of the Navy's Information Technology (IT) vision. "These initiatives [NMCI and Task Force Web] are critical components to the Department of Navy's vision of a network-centric force," Edmonds says. "The ability to access, process, and disseminate information rapidly and securely has a direct impact on force readiness. The old axiom that 'knowledge is power' was never more true and that is what the Information Strike Force is committed to provide to our naval forces – the power of information."[4] Al Edmonds is the president of the EDS federal government division, which is tasked with implementing NMCI.

[3] *Government maintained non-public website: https://tfw-opensource.spawar.navy.mil/servlet/portal/?escmd=startup.*

[4] *McHale, John. "Navy Marine Corps Intranet Goes Online at its First Military Base"* Military & Aerospace Electronics *Feb. 2002.*

4.6.1 Conclusions and Recommendations

Task Force Web provides the Command and Control user the ability to access multiple web applications through a single point of entry. There is a significant benefit to this level of accessibility. It further promotes a future vision of a web services architecture that will allow greater integration of data from multiple data sources. The benefits to this n-tier web services approach to the end user are significant: provides access to tactical and logistics data that improves decision making and business efficiency, decreases the total cost of ownership of redundant systems, and provides ease-of-use to the user. In addition, the integration of a Single Sign On with a single network account will provide an end-user greater ease of use. It also promotes a network-centric computing approach wherein a C2 user can login to any desktop with a browser and access web applications. This flexibility is a desirable feature for the warfighter. The goals of the Task Force Web initiative are worth achieving and will have significant benefit once they are fully implemented.

4.7 DoD Metadata Registry and Clearinghouse

DoD Metadata Registry and Clearinghouse: The DoD Metadata Registry and Clearinghouse at http://diides.ncr.disa.mil registers information about XML for registered communities of interest (COIs). A typical COI would be command and control. The information about XML that is registered includes XML Schema documents, and example XML instance documents for those schemas. At the time that this information is registered, metadata that describes it is supplied as part of the registration process.

The DoD Metadata Registry and Clearinghouse has both a browser based interface and a REST-based web interface. Authorized users can query the registered information based on the metadata that is associated with it, which was supplied at registration time.

> For Web services, the DoD Metadata Registry and Clearinghouse could serve as a stable repository for other kinds of data that is at the same level of abstraction as XML Schema documents. An important example of this other kind of data would be WSDL documents, which themselves include XML Schema documents (which may in fact be registered in the DoD Metadata Registry and Clearinghouse). Another example of similar kind of data that could be registered is taxonomies and ontologies encoded in OWL.

5 Special Considerations for C2 Programs

Command and Control (C2) applications have some common characteristics that may be impacted by Web Services. Among these are Security, Messaging, and common applications areas such as geospatial data representation and presentation, symbology, sensor collection management, and sensor processing. Web user interfaces are also becoming more important for user interaction with C2 systems, and Web Portal technologies and standards are relevant.

5.1 Security (Restricted Access, and Multi-Level Security)

The following concepts are relevant to Web Services security:

- Identification and Authentication. The *identification* process enables recognition of an entity (subject or object) by a computer system. An *authentication* procedure establishes the validity of a claimed identity.
- Authorization. The *authorization* process determines whether an authenticated identity may have access to particular computer system resources or data.
- Access Control. *Access control* is the process to limit access to the resources of a system only to authorized identities, processes, or other systems in a network. There are two strategies for defining an access control policy: *need-to-know* and *need-to-hide*. Need-to-know restricts access only to those resources or data for which an identity is authorized. Need-to-hide restricts access from those resources or data for which an identity is not authorized. *Department of Defense Directive 8500.1 Information Assurance (IA)* states that the access to DoD information systems shall be need-to-know.
- Detect and Response. *Detect and response* refers to the capability for the rapid detection of, and reaction to, intrusions. This capability generally has a *fusion* aspect so one incident can be viewed in relation to others. This allows for the identification of potential activity patterns or new developments.

The following standards, practices, and implementations apply to Web Services security:

- Identification. Identification is generally done in a web services environment by the use of unique machine-readable user names. A standard for password usage is *FIPS PUB 112, Password Usage*, 30 May 1985. Additionally, a one-time password standard for login authentication that is secure against passive attacks based on replaying captured reusable passwords is *IETF RFC 2289, A One-Time Password System*, February 1998.
- Authentication Servers. Authentication servers use security measures to establish the validity of a transmission, message, or originator. A standard for user authentication in a distributed computing environment is Kerberos (*IETF RFC 1510, The Kerberos Network Authentication Service, Version 5*, 10 September 1993); an emerging standard for remote dial-in is RADIUS (*IETF RFC 2138, Remote Authentication Dial In User Service (RADIUS)*, April 1997).

- Access Control. Access control encompasses authorization implementations that use list-based mechanisms to determine privileges within a system. Access control mechanisms may be categorized as either *capabilities-based* or *resource-based*, or *hybrid*.
    - In a *capabilities-based* scheme, the list is associated with the user, and contains capabilities. A user by default has no capabilities, and is granted capabilities by having them added to his list.
    - In a *resource-based* scheme, the list is associated with the resource, and contains user references. Users on the list have access to that resource. This is the canonical Access Control List (ACL).
    - There are many possible *hybrid* schemes. For instance, if the system treats the ACL mechanism itself as a grantable capability, an appropriately granted user may override the restriction of the ACL. In another variant, the ACL for a resource may have, in addition to the user reference, a list of capabilities automatically granted to that user, but only applicable to that resource.
- Multi Level Security (MLS). A MLS system is an access control implementation that compartmentalizes user actions according to specific *security labels*. Security labels are associated with each user, process, resource, and data object. Security labels contain one or a combination of two elements: a *sensitivity level* and (in some security schemes) an *integrity grade*. If the security label has only a sensitivity level, then a test of that is all that is required to determine access. If the security label has both a sensitivity level and an integrity grade, the both tests need to be satisfied for access, and any case of failure prohibits access.
    - *Sensitivity level* defines the secretness or classification of files and resources and the clearance level of users. Sensitivity levels are hierarchical: the user or process must have a sensitivity level at or above the sensitivity level of the resource or data in order to view or access it. An example of sensitivity level is the US Government classification levels (i.e. “top_secret”, “secret”, “confidential”, “unclassified”,). An individual with a “top_secret” security level can view a “confidential” application, but an individual with a “confidential” security level cannot view a “top _secret” application. In addition to sensitivity level, a non-hierarchical, non-exclusive *sensitivity category* can be assigned to compartmentalize users, processes, resources, and data. One example of the use of sensitivity categories may be the geographic region of origin (e.g. “north_america,” “south_america,” “asia,” “africa,” and “europe”). A user or process must have the sensitivity category (or categories) of the resource or data, in order to view or access it.
    - While the sensitivity levels identify whether a user is cleared to view certain information, *integrity grades* identify whether data is reliable enough for a specific user to see or access. Integrity grades are inverse-hierarchical: the user or process must have an integrity grade at or below the integrity grade of the resource or data in order to view or access it. An

example of integrity grades is executable software in an operating system environment (e.g. "verified_safe," "presumed_safe," "unknown_safe," "known_virus"); a "presumed_safe" program could see and execute a "verified_safe" program, but not an "unknown_safe" program (thus preventing a more-trusted behavior from executing a less-trusted behavior). Another example is the use of integrity grade to classify the reliability of information and sources (e.g. "confirmed_fact," "observation," "guess"); in this example, a document that has an "observation" integrity grade could not include data sources that have "guess" grade, but could include a "confirmed_fact" (thereby maintaining the integrity of the document's information). Just like with sensitivity categories, a non-hierarchical, nonexclusive *integrity category* can be assigned to compartmentalize users, processes, resources, and data. For example, information can be categorized by ownership (e.g. "government," "commercial," "academic," "scientific," and "public"). A user or process must have the integrity category (or categories) of the resource or data, in order to view or access it.

- Secure Web Browsing. This service identifies the protocol used to provide communications privacy over a network. The protocol allows applications to communicate in a way designed to prevent eavesdropping, tampering, or message forgery. Web services provide abilities for navigation and data transport across the Internet. The protocol encapsulates various higher-level protocols and is application independent.

  According to the JTA, Web browsers and web servers must first attempt to use TLS, then use SSL 3.0 if TLS is not supported. It is expected that SSL 3.0 will not be supported in the future. The following standards are both mandated for securing the communications of web browsers and web servers: *Secure Sockets Layer (SSL) Protocol, Version 3.0*, 18 November 1996 and *IETF RFC 2246, The Transport Layer Security (TLS) Protocol Version 1.0*, January 1999.

5.1.1 Information Assurance

[Sources:

1. *Information Assurance Technical Framework, Release 3.1*, September 2002, National Security Agency http://www.iatf.net
2. *Department of Defense Directive 8500.1 Information Assurance (IA)*, 24 October 2002
3. *Department of Defense Instruction 8500.2 Information Assurance (IA) Implementation*, 6 February 2003]

Information assurance (IA) is the set of policies and procedures by which sensitive data within information system is protected, and if attacked, mitigated and recovered. The DoD IA policy is defined in *Department of Defense Directive 8500.1 Information Assurance (IA)*, which states that all DoD information systems shall maintain an appropriate level of confidentiality, integrity, authentication, non-repudiation, and availability that reflect a balance among:

- The importance and sensitivity of the information and information assets.
- Documented threats and vulnerabilities.
- The trustworthiness of users and interconnecting systems.
- The impact of impairment or destruction to the DoD information system.
- Cost effectiveness.

For IA purposes all DoD information systems shall be organized and managed in four categories:

1. Automated information system (AIS) applications.
2. Enclaves (which include networks).
3. Outsourced IT-based processes.
4. Platform IT interconnections.

IA attempts to limit damage and recover rapidly from attack. There are five classes of attacks:

1. Passive. *Passive attacks* include traffic analysis, monitoring of unprotected communications, decrypting weakly encrypted traffic, and capturing authentication information (e.g., passwords). Passive intercept of network operations can give adversaries indications and warnings of impending actions. Passive attacks can result in the disclosure of information or data files to an attacker without the consent or knowledge of the user.
2. Active. *Active attacks* include attempts to circumvent or break protection features, introduce malicious code, or steal or modify information. These include attacks mounted against a network backbone, exploitation of information in transit, electronic penetrations into an enclave, or attacks on an authorized remote user when attempting to connect to an enclave. Active attacks can result in the disclosure or dissemination of data files, denial of service, or modification of data.
3. Close-in. *Close-in attack* is where an unauthorized individual is in physical close proximity to networks, systems, or facilities for the purpose of modifying, gathering, or denying access to information. Close proximity is achieved through surreptitious entry, open access, or both.
4. Insider. *Insider attacks* can be malicious or non-malicious. Malicious insiders have the intent to eavesdrop, steal or damage information, use information in a fraudulent manner, or deny access to other authorized users. Non-malicious attacks typically result from carelessness, lack of knowledge, or intentionally circumventing security for non-malicious reasons such as to get the job done.
5. Distribution. *Distribution attacks* focus on the malicious modification of hardware or software at the factory or during distribution. These attacks can introduce malicious code into a product such as a back door to gain unauthorized access to information or a system function at a later date.

To achieve an effective IA posture the DoD had defined a strategy called *Defense-in-Depth*: organizations address IA needs with *people* executing *operations* supported by *technology*. The following outlines the principle elements and relevant aspects of Defense-in-Depth:

- People
    1. Policies and Procedures.
    2. Training and Awareness.
    3. Physical security.
    4. Personnel security.
    5. System security administration.
    6. Facilities Countermeasures.
- Operations
    1. Security policy.
    2. Certification and accreditation.
    3. Readiness assessments.
    4. Security management.
    5. Key management.
    6. Attack sensing and warning response.
    7. Recovery and reconstitution.
- Technology
    1. IA Architecture framework areas.
    2. IA criteria (security, interoperability, and PKI).
    3. Acquisition integration of evaluated products.
    4. System risk assessments.

The implementation of Defense-in-Depth is described in *Department of Defense Instruction 8500.2 Information Assurance (IA) Implementation*, 6 February 2003.

5.1.2 Joint and Coalition Systems Interoperability

[Source: *Department of Defense Joint Technical Architecture v5.0*, 4 April 2003.]

Joint and Coalition Systems Interoperability are addressed in the Joint Technical Architecture (JTA). The JTA objective is interoperability in the following areas:

- Within a Joint Task Force/Combatant Command Area of Responsibility (AOR).
- Across Combatant Command AOR boundaries.
- Between strategic and tactical systems.

- Within and across Services and Agencies.
- From the battlefield to the sustaining base.
- Among U.S., Allied, and Coalition forces.
- Across current and future systems.

The JTA discusses Joint interoperability in much greater detail than Coalition interoperability. The JTA emphasizes facilitating interoperability in joint and coalition force operations by mandating standards and guidelines for DoD systems.

Both the Joint and Coalition environments require that some members have specific or limited access to information or services. For example, a Joint exercise will require that certain elements must receive certain kinds of information (e.g. weather data or geographic troop dispositions), while others may receive it, and still others may not access it at all.

Similarly, a certain Coalition member may wish to hide the identity of an intelligence asset from all other Coalition members, and may share the raw intelligence reports from that asset with certain Coalition members, and share a sanitized digest of that report with a greater circle of Coalition members.

Each of these examples illustrates a requirement for both Access Control and Multi Level Security.

5.1.3 Recommendations for Security and Information Assurance

The following specification is recommended for addressing Web Services security requirements:

- *Security Assertion Markup Language (SAML), Version 1.1*, September 2003, OASIS Security Services Technical Committee.

SAML resolves identification, authentication, authorization, attributes, security authorities, and exchange of authentication and authorization information across security domains (including identity management and single sign-on). SAML incorporates industry-standard protocols and messaging frameworks, such as XML Signature, XML Encryption, and SOAP. SAML currently defines one binding, to SOAP over HTTP.

SAML can also be integrated with existing system security services to address Multi Level Security (MLS) issues, and can be used to implement and extend various Access Control schemes.

While people and policies are critical to IA, the technological foundation of IA is in a trusted infrastructure. This necessarily includes infrastructure that is external to the control of the IA administrator, so the recommended specifications must allow for such a heterogeneous or even adversarial environment.

To address technical IA issues, the OASIS Web Services Security (WS-Security) specifications for SOAP Message Security, Username Token Profile, and X.509 Token Profile, WS-SecureConversation, and WS-Trust are recommended. In concert with SAML, these specifications provide identity and authentication, and message integrity and confidentiality, even across an adversarial environment. OASIS Extensible Access Control Markup

Language (XACML) or Web Services Policy Framework (WS-Policy) can additionally be implemented for authorization and access control.

5.2 DoD Messaging Standards

[Source: *Department of Defense Joint Technical Architecture v5.0*, 4 April 2003.]

The Joint Technical Architecture defines two classes of tactical information exchange standards: *bit-oriented* and *character-based*.

Bit-oriented fixed and variable formatted Tactical Data Link (TDL) standards allow real or near real-time tactical digital information exchange among air, ground, and maritime components of U.S., NATO, other allies, and friendly nations. Among the bit-oriented message formats are:

- LINK 16. Link 16 is a secure, jam resistant, nodeless data link that uses the Joint Tactical Information Distribution System (JTIDS)/Multifunctional Information Distribution System (MIDS) time division multiple access (TDMA) protocols, conventions, and fixed message formats. Link 16 provides for the real/near real-time exchange of air, space, surface, subsurface, and ground tracks, and orders and commands among participating units. MIL-STD-6016B defines the Link 16 message set, minimum implementation, data forwarding, and system implementation specifications, and a common data element dictionary (DED). The following standard is mandated for bit-oriented formatted messages: *MIL-STD-6016B, Tactical Digital Information Link (TADIL) J Message Standard*, 1 August 2002. In a NATO environment, the following standard is mandated: *STANAG 5516, Edition 2, Tactical Data Exchange LINK 16*, Ratified 10 November 1998.
- Variable Message Format. Variable Message Format (VMF) is the DoD mandated standard for fire support information digital entry device exchange over tactical broadcast communications systems. The use of VMF has been extended to all war fighting functional areas. The VMF Technical Interface Design Plan (Test Edition) (TIDP-TE) defines the VMF message set and data element dictionary (DED). VMF minimum implementation and data forwarding requirements are under development. The standard for VMF is defined in *Variable Message Format (VMF), Technical Interface Design Plan (Test Edition) Reissue 5*, 18 January 2002.
- LINK22. Utilizing J-series messages and data elements, Link 22 uses an improved high frequency (HF) and ultra-high frequency (UHF) multimedia transmission scheme. The link uses Time Division Multiple Access (TDMA) protocols, is capable of multi-netting, and provides 300 nautical mile coverage using HF and line-of-sight connectivity using UHF. *STANAG 5522, Edition 1, Tactical Data Exchange LINK 22* (September 2001) is the Multinational Group (MG) agreed Configuration Management (CM) baseline document as of 15 September 1995. It is distributed as *ADSIA (DKWG)-RCU-C-74-95*.

Character based information standards, provide common, human-readable, and media independent messages used for planning and execution in joint and combined operations among U.S. forces, NATO, other allies, and friendly nations. The following formats exemplify these kinds of messaging standards:

- United States Message Text Format. United States Message Text Format (USMTF) messages are jointly agreed, fixed-format, character-oriented messages that are human-readable and machine-processable. USMTFs are the mandatory standard for record messages when communicating with the Joint Staff, Combatant Commands, and Service Components. The following Character-Based Formatted standard for USMTF messages is mandated: *MIL-STD-6040, United States Message Text Format (USMTF)*, 31 March 2002.
- Over the Horizon Targeting (OTH-T) GOLD. This format is widely used in Navy and other systems for distributing tactically relevant information to manage a common operational picture but also for other forms of communication (including operator notes, or OPNOTES).

Secure Messaging. This service applies to the use of security implementations for the Defense Messaging System (DMS), the access control capabilities for communications with Allied partners, and for e-mail.

For systems required to interface with the Defense Message System, DMS Release 3.0, for Organizational messaging, the following standard is mandated:

- FORTEZZA Interface Control Document, Revision P1.5, 22 December 1994.

For DoD message systems required to process both unclassified and classified organizational messages using DMS Release 3.0, the following messaging security protocol is mandated:

- ACP-120, Allied Communications Publication 120, Common Security Protocol (CSP), Rev A, 7 May 1998.

To support the access control capabilities of ACP 120, the following security label standards are mandated:

- ITU-T Recommendation X.411 (1999)/ISO/IEC 10021-4:1999, Information Technology Open Systems Interconnection Message Handling Systems (MHS) Message Transfer System: Abstract Service Definition Procedures.
- ITU-T Recommendation X.509 (2000)/ISO/IEC 9594-8:2001, Information Technology Open Systems Interconnection The Director: Public Key and Attribute Certificate Frameworks, 2001, with Technical Corrigendum 1:2002, and Technical Corrigendum 2:2002.
- ITU-T Recommendation X.481 (2000)/ISO/IEC 15816-12:2000, Information Technology Security Techniques Security Information Objects for Access Control.
- SDN.706, X.509 Certificate and Certificate Revocation List Profiles and Certification Path Processing Rules, Revision D, 12 May 1999.
- SDN.801, Access Control Concept and Mechanisms, Revision C, 12 May 1999.

The Secure/Multipurpose Internet Mail Extensions (S/MIME) v3 protocol suite provides application layer privacy, integrity, and non-repudiation (proof of origin) security services for messaging (e-mail). Three IETF RFCs (RFC 2630, RFC 2632, and RFC 2633) provide the above listed core security services. For individual messages that use certificates issued by the DoD PKI to protect unclassified sensitive information or sensitive information on system high networks the following standards are mandated:

- IETF RFC 2630, Cryptographic Message Syntax, June 1999.
- IETF RFC 2632, S/MIME Version 3 Certificate Handling, June 1999.
- IETF RFC 2633, S/MIME Version 3 Message Specification, June 1999.

5.2.1 Recommendations for DoD Messaging

It is recommended that Web Services employ XML-based messaging, appropriate to the application. The benefits of XML-based messaging are being recognized by the DoD and allied military organizations. The universally accepted notion is that all messaging systems in the future (including current legacy systems) will at least have an XML message payload, and most likely be communicated over an XML-based service.

It is recommended that the following specification be adopted for message structure:

- *Simple Object Access Protocol (SOAP) 1.2, W3C Recommendation*, 24 June 2003.

In general both character and binary message data can be expressed as a SOAP message with a formatted attachments. The SOAP message can be transmitted using HTTP or another protocol as appropriate, for example SMTP or BEEP. Reliable messaging specifications (e.g. WS-Reliability, ebMS, WS-ReliableMessaging, and WS-Acknowledgement) are still under development.

Efforts are underway at DoD and other national and international bodies to migrate legacy message formats XML extensions. For instance, XML-MTF was approved by United States Message Text Format (USMTF) Configuration Control Board (CCB) in February, 2001, to be included in published USMTF 2002 baseline, which becomes effective in March, 2002. XML-MTF was formally ratified by NATO member nations in March 2001, to be included in Allied Data Publication 3 (ADatP-3) baseline 12, to be published in Fall, 2001.

5.3 C2 Relevant Domain Standards

5.3.1 Geospatial Standards

5.3.1.1 The Open GIS Consortium (OGC)

The OpenGIS Consortium is a non-profit organization whose goal is to promote interoperability between web services. To accomplish this, OGC defines and publishes standards for geographic data services. All of these standards are freely available to the general public from the OGC web site, http://www.opengis.org. The OGC services discussed here all make use of XML for data transport and use HTTP as their distributed connection mechanism.

OGC provides only the specifications for web services. To get a working OGC web service, one must download, buy, or write some software. (Fortunately, various implementations of these specifications are available both in open-source and as shrinked wrapped products from vendors.)

If Web Services architecture were to make use of the OGC standards, there would many benefits to the developer and user of C2 applications:

- Since all OGC standards use well-defined protocols and formats, the user can potentially see many types of data on the same map that were previously visible only separately.
- Because server interfaces are carefully defined, new servers can be deployed without virtually no impact on the other existing servers and clients.

Development time is decreased for new clients and servers since a large body of freely available open-source code exists for implementing OGC services and clients.

5.3.1.2 Open GIS Consortium Web Standards

The Web Services Standards Analysis Report covered several OGC standards relevant to C2. To summarize:

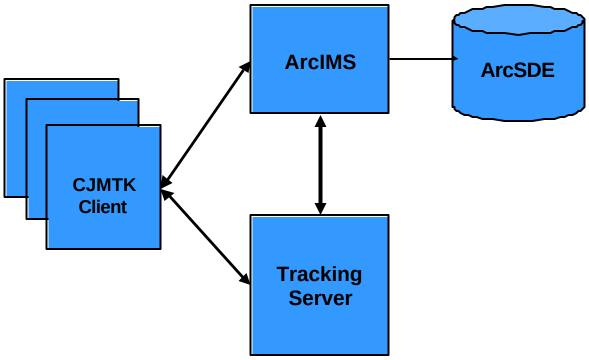
- WMS - One of the earliest standards to arise from the efforts of OGC was the specification of a Web Mapping Service, or WMS. Such a service provides rendered map imagery (typically as JPG, GIF, or PNG files) of any data that the server wishes to provide. This might include features (roads, rivers, power lines) or imagery (aerial photography, satellite photos, weather imagery).
- WFS - For applications that display maps to a user, a WMS is sufficient. However, some applications need access to the underlying data for a given set of features. The Web Feature Server, or WFS, provides a mechanism to query for data as objects (called “features”) and collections of objects (“feature collections”). Each feature can have geometry (in GML, described next), and an application-defined, XML Schema-based, set of feature data as XML content (e.g., attributes about the object). In C2, a “track” could be a feature, the location and perhaps area of uncertainty and/or movement might be described as GML, and other information would be associated with that geometry such as Date Time Group (DTG), Track ID, and other identifying or sensor-supported information.
- GML - In order for the WFS to provide the underlying data for a set of features, a transport mechanism had to be developed. The Geographic Markup Language, GML, was developed to suit this purpose. GML is an XML-based standard that provides an application with a set of geographic primitives, such as points, lines and polygons, for defining the attributes and geographic geometry of a feature.

These standards have been applied in existing C2 applications (many of them described elsewhere in this document). We recommend the DoD officially adopt these OGC standards as the basis for implementation of web-based geospatial services, and the basis for XML representation of geospatial data.

5.3.1.3 CJMTK

The Commercial Joint Mapping ToolKit (CJMTK) has been introduced by a successful proposal by Northrup Grumman Information Technology (NGIT) and their primary subcontractor Environment Systems Research Institute (ESRI). The intent is to provide a common and commercially maintained API for visualization, persistence and serving of geographic data. The CJMTK consists of several core tools and technologies. The complete scope and capabilities of these tools is too numerous to mention in this document, so the focus of this section will be focus on the particular tools and capabilities of these tools that are relevant to web services and related technologies. Discussion of how these tools can provide web services, how web services can be consumed by these components will be covered. Components exist for many kinds applications including spatial data persistence, spatial data application server, thin client components, thick client components and messaging.

The application server role is filled by ArcIMS™. ArcIMS is capable of consuming and publishing data from a variety of sources. These sources include raster images, shapefiles, CAD drawings, other ArcIMS servers, ArcSDE data as well as many other data types. In a system with much diverse data and many existing applications ArcSDE is of particular importance. Data published directly by ArcSDE does not conform to any open specifications or widely accepted open standards. However, using special connectors data within ArcIMS can be published in a form compliant with the OGC specifications for Web Map Servers (WMS) version 1.1.1 and Web Feature Servers (WFS) version 1.0.0 so that clients already written to these specifications have access to data published by ArcIMS. This significantly broadens the potential client base and integration possibilities for ArcIMS. The diagram below shows a typical arrangement of the ESRI components mentioned above.



Clients that wish to retrieve data directly from an ArcIMS server must use a particular query syntax written in a proprietary language called ArcXML. ArcXML is an XML-based standard similar in spirit to GML that provides mechanisms for querying and displaying geographic information. There are several ways to specify constraints on queries constructed with the language, including a spatial areas and particular attributes of data items with syntax reminiscent of SQL. With the wrappers for WMS and WFS compliance applied their respective query capabilities may be utilized as an alternative to using the ArcIMS proprietary ArcXML syntax.

Relational databases, such as Informix, Oracle, or Microsoft SQL Server, provide efficient mechanisms for querying and storing large quantities of data. ESRI's ArcSDE enables these databases to store geo-spatial information and provides a mechanism for ArcIMS to retrieve and display this data. On its own, ArcSDE is not a web service. It provides a set of C++ and Java APIs for accessing the data. But combined with ArcIMS, it can provide an efficient mechanism for querying and displaying geospatial data. The APIs exposed by ArcSDE allow arbitrary data to be stored, including data with rich geometric/geographic representation. This provides existing applications the means to easily and dynamically publish data that will ultimately be exposed via ArcIMS or consumed directly from ArcSDE by other servers or thick client applications. It is at this level that applications have already been built to import data from any OGC compliant WFS sever into the CJMTK suite of components.

Currently ArcIMS does not provided a means to directly consume data from other standard OGC WFS components. Polexis incorporated currently provides Web Feature Loader (WFL) that is capable of transferring arbitrary data from other WFS implementations, as well as data from XIS aware data source interface components, directly into ArcSDE. Integration at this level is desirable because it exposes the data to the largest number of CJMTK aware components. With this infrastructure enabled it is possible share data across many chained servers

Many web service clients have a need to be kept up to date with real time changes in data. When many clients are involved an efficient means of keeping these clients up to date is required. ESRI's tracking server fills this messaging role by building on top of Java Messaging Service (JMS) APIs. Notification of updates to ArcIMS, or other components that publish events to the tracking server, are sent to interested clients prompting then to update to local view of the data with those changes. This is preferable to requiring clients with dynamic update needs to implement their own web service components for callbacks because it is much simpler to utilize in the client programs.

5.3.2 Symbology

C2 systems use symbology (icons, line styles, fill styles, etc.) extensively as a way to quickly convey geospatially-referenced information to C2 users. These symbols are used to indicate the positions of friendly and enemy forces, lines of advance, strategically meaningful regions (places that forces should or should not be), targets, etc. Web Services must not only convey information about C2 objects, but often must also convey metadata about how strategically- or tactically-relevant information should be rendered. Standard symbology is extremely important in order to ensure that a situational awareness picture always conveys the same meaning to all users.

5.3.2.1 SVG

The Scalable Vector Graphics, or SVG, standard provides explicit control for the rendering of data in a web environment. SVG is a W3C recommendation that allows content authors to create complex graphical renderings that are zoomable, animated, and can have complex interactions with the user. SVG allows the content provider to make use of CSS and XSL to control the rendering of the graphics.

5.3.2.2 SLD

When using web services that utilize Open GIS Consortium interfaces (such as those described above), the display of features can be controlled using documents written using the "Styled Layer Descriptor", or SLD, specification. As with all other OGC specifications, SLD is XML based. It provides mechanisms for defining rendering characteristics such as line width, color, icons for point symbols, etc.

5.3.2.3 MIL-STD 2525B

For the DoD, standards such as SLD and SVG have significant limitations. They are designed to be very general, and while you may (or may not) be able to use these standards to generate desired symbology, complicated symbology and styling can be difficult.

The DoD has created a symbology standard called Military Standard 2525B (MIL-STD-2525B), which defines an extensive set of single-point symbols (icons with text and graphic annotations), boundary lines, multi-point lines, and other indicators (e.g., bridges). This standard also defines fill patterns (e.g., mine fields) and line styles (e.g., razor wire), with very explicit expectations for how these styles should be rendered.

The standard does provide ways to express the desired symbology through various codes, but no XML standard has been proposed to cover the entire specification, and because this standard is tied to such a narrow domain (DoD), it is unlikely a commercial standard will emerge.

We recommend that the DoD define an XML adaptation of MIL-STD-2525B that is based on commercial standards such as GML and SLD. A standard schema for symbology would be critical to supporting net-centric C2 computing architectures and system interoperability.

5.3.3 Sensor Collection Management

The OpenGIS Consortium (OGC) is developing standards for sensor collection management. SensorML is a language for describing sensors and the platforms that carry them. SensorML covers both dynamic satellite and UAV mounted sensors, and static ground or ocean based sensors. SensorML is not a language for describing sensor data.

OGC is also developing Observations and Measurements (O&M), which is a language for describing sensor data that is based on OGC's GML3 language. O&M can describe point, vector, coverage, or time series data.

5.3.4 Web Portals

Portals provide a convenient means to integrate disparate web applications, web services and content into a single user interface. The goal is to present a wide variety of information in a summarized and easily accessible form. Users can then drill-down into more detailed information as needed. This allows a wide variety of content and services to be integrated into a single web application, accessible from a single entry point. This integration typically leads to custom code that is used to obtain, format or summarize data from many different sources, using a wide variety of technologies. Until recently the lack of a standard portal component API (commonly called portlets) has a prevented this code from being portable between portal products. In addition, there was no standard way to interact with web services that provided UI capabilities, or a standard way to write web services that provided their own UI. Recent specifications have addressed these issues.

With JSR-168 the Java Community Process has created a standard API for developing and deploying portlet components. The standard provides mechanisms for controlling "window like" portlet behaviors as well as several modes in which portlets can operate. Several implementations of this standard have already emerged in both commercial and free software products. This provides benefits to portals similar to those provided by Servlets to web applications. These components can be easily added to and removed from portals, as well as purchased from vendors other than the provider of the portal product in which they will be deployed. Each portlet is also capable of providing its own security constraints based on user and role information, as well as providing for the confidentiality of data provided by the portlet. The ability of each portlet to manage their security is of particular importance for portals in the C2 domain. It frees the developers creating the portal itself from being concerned with the security needs of individual components.

An additional open specification called Web Services for Remote Portlets (WSRP) allows portlets to expose their capabilities as a web service. As a web service their content is markup that represents a UI to a particular piece of functionality. Portal containers can then host the portlets by using a generic local proxy for remote portlets. This can greatly simplify the deployment of portals that need diverse capabilities by allowing them to aggregate functionality from servers in many locations without writing custom components to manage the interactions. WSRP (unlike JSR-168 which is specific to Java™) does not place any restriction on the tools or technologies used to implement a portlet component. As such it is entirely possible that remote portlets could be written and hosted in environments other than J2EE application servers.

The combined capabilities represented by these specifications allow portals to be created more easily by utilizing pre-built components hosted both locally and remotely. Having a large set of these components readily available and exposing a wide variety of data sources allows users to quickly adapt to changing conditions. The leveraging of JSR-168 and WSRP, and the component market they are likely to create holds potential to significantly speed development and reduce development cost. These benefits are widely recognized in industry and are arguably the motivating factors driving the development of specifications by standards groups.

6 Case Studies

6.1 Extensible Tactical C4I Framework (XTCF)

XTCF provides a common information management framework that enables multiple data sources, transformation services, analysis tools and data management services to cooperate in producing a common tactical information network service. As such, XTCF represents an initial version of GIG services, for the JC2 community of interest. This framework supports runtime integration of new data sources, new data storage agents, new correlation services, new information distribution services and new information query services.

6.1.1 Architecture and Standards

6.1.1.1 Architecture

XTCF provides a Service Oriented Architecture (SOA) augmented with publish and subscribe capabilities. XTCF supports either JMS messaging or SOAP Web Services over HTTP. This flexibility makes XTCF accessible to a broader range of service providers and service consumers.

The XTCF design encourages the use of asynchronous messaging and asynchronous behavior in service transactions. Notifications and alerts, including sensor events, are inherently asynchronous. But even traditionally synchronous transactions, such as request/response, can be implemented as a sequence of asynchronous messages, which is the default implementation for XTCF components. This design, combined with reliable messaging, can support disconnected operations.

6.1.1.2 Standards

This section provides an overview of standards that are relevant to XTCF.

6.1.1.2.1 Extensible Markup Language (XML)

Data exchanged in XTCF is expressible in XML, providing a well-known and easy to use format for interfacing systems. A common data model in an XML Schema Document (XSD) provides consistency in how the XML tags should be interpreted.

6.1.1.2.2 Web Services Description Language (WSDL)

Web Services Description Language (WSDL) is used for the services provided by XTCF components, and the services on which they depend, to express the operations provided, the input and output parameters for those operations, and one or more protocol bindings (ways to connect).

6.1.1.2.3 Universal Description, Discovery and Integration (UDDI)

UDDI provides a registry of web services for advertisement, discovery, and integration purposes. UDDI maintains a directory for representing business entities, their relationships, and the services they provide. UDDI supports discovery of available web services by providing searches by name, identifier, category, or implemented specification.

XTCF uses UDDI to keep track of the web services that registered components offer, and to provide discovery of those services in response to a request. A requestor can discover services provided within a given domain, or can search outside the domain.

6.1.1.2.4 Simple Object Access Protocol (SOAP)

XTCF core services and services supporting plug-ins are accessible via a SOAP interface. The advantage of the SOAP interface is its universal support, thus providing the widest range of interoperability. It is particularly well suited to request-response interactions.

6.1.1.2.5 Java Message Service (JMS)

In XTCF, a messaging system is used to provide the necessary asynchronous exchange of messages between plug-in components. By employing the JMS API to interact with the messaging system, a plug-in developer can choose the supplying vendor on the basis of varying criteria for each XTCF domain. A free implementation such as JBoss can be used in less demanding environments (such as development) and a more robust implementation such as SonicMQ or WebLogic can be used for a fielded system requiring more performance and stability.

6.1.1.2.6 Lightweight Directory Access Protocol (LDAP)

XTCF uses LDAP to locate and access the message broker and core services (registration, discovery, statusing) for a given domain. LDAP is also be used to keep track of users and their associated groups and roles

6.1.2 Development Techniques

XTCF provides support for the automatic development of Web services. Tools in the XTCF SDK can be used to automatically generate the WSDL for an XTCF plug-in. The java source for the Web service that implements that WSDL is also generated, as is the source code for a client to that service.

6.2 Global Combat Support System (GCSS) Web Portal

Global Combat Support Systems (GCSS) provides information interoperability across combat support and command and control functions. Existing data sources and system components are integrated at an enterprise level on a J2EE n-tier architecture. The primary objective for the GCSS is to provide a cohesive Combat Support (CS) picture of the battlespace to the warfighter, serving the Combatant Commanders (CCs) and their established Joint Task Forces (JTF). The commanders are supplied with read-only access to authoritative CS information from various CS databases that store combat support status. The mission of GCSS (CC/JTF) is to provide end-to-end information interoperability across both CS and Command and Control (C2) functions in support of the CC and JTF Combat Support requirements. The GCSS Concept and GCSS (CC/JTF) bridge the gap between C2 and CS logistics and allow successful execution of missions.

6.2.1 The GEOLOC LookupTable

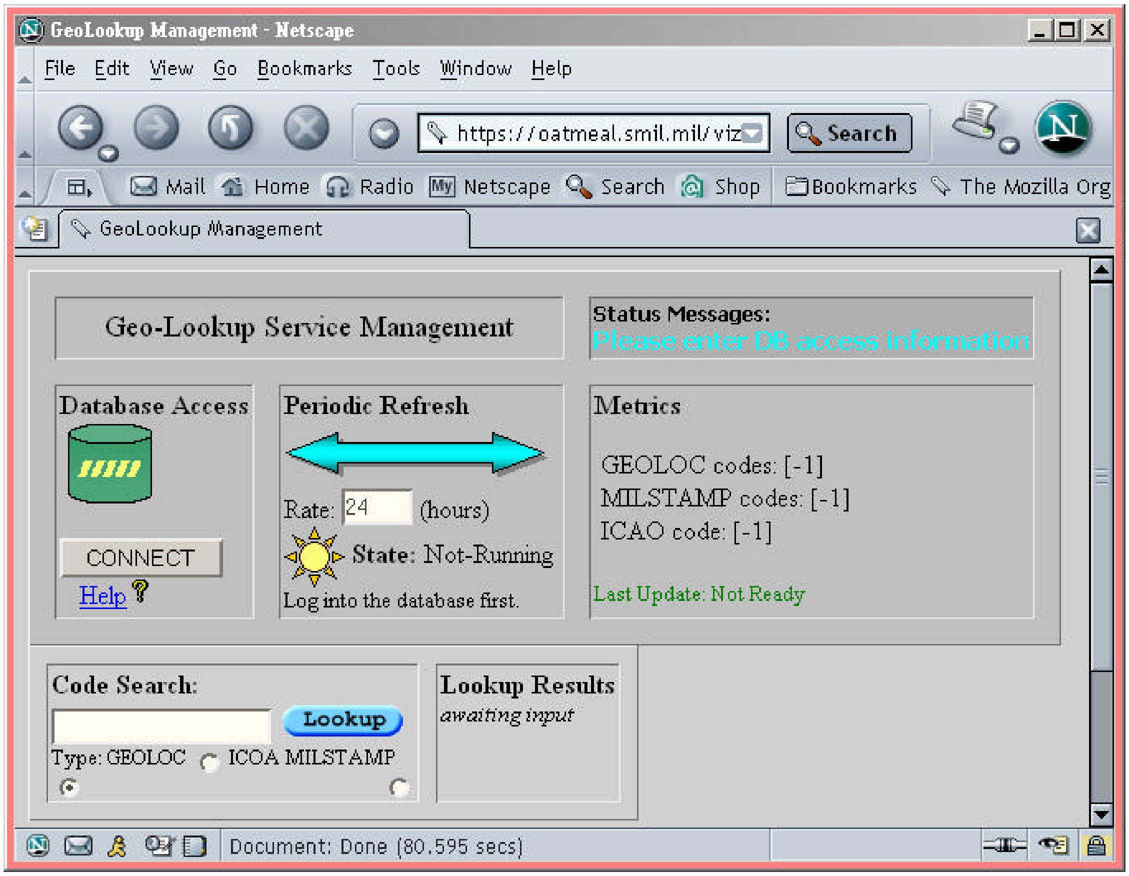
The GEOLOC LookupTable is an Enterprise Java Bean (EJB) designed to mitigate performance issues during the resolution of latitude and longitude values (for several types of codes). LookupTable development was motivated by a requirement to provide an API to facilitate the lookup process. Using the mil.disa.gcss.util.lookup.geolocation package is pretty simple.

The necessary LookupTable APIs belongs to the mil.disa.gcss.util.lookup.geolocation.LookupTable class. They are:

- LookupTable::getLookupTable – returns the LookupTable
- LookupTable::lookup(String code, String value) – returns an array of latitude and longitude values for the given code/value pair.

A Java Servlet initialization process automates the creation of the lookup table. However, since this capability is general, it might also be used outside of the GCSS (Web Application) environment.

The LookupTable InfoBean class has a render method that responds to HttpServletRequests that contain domain-specific parameters like *NAME_KEY* , *PASSWORD_KEY*, .*FILE_LOCATION_KEY, etc …*


GeoLookup Management - Netscape
File Edit View Go Bookmarks Tools Window Help
Search
Mail Home Radio Netscape Search Shop Bookmarks The Mozilla Org
GeoLookup Management
Geo-Lookup Service Management
Database Access
CONNECT
Help
Periodic Refresh
Metrics
Code Search:
Lookup
Lookup Results

https://oatmeal.smil.mil/viz

Status Messages:
Please enter DB access information

Rate: 24 (hours)
State: Not-Running
Log into the database first.

GEOLOC codes: [-1]
MILSTAMP codes: [-1]
ICAO code: [-1]

Last Update: Not Ready

Type: GEOLOC ● ICOA ○ MILSTAMP ○

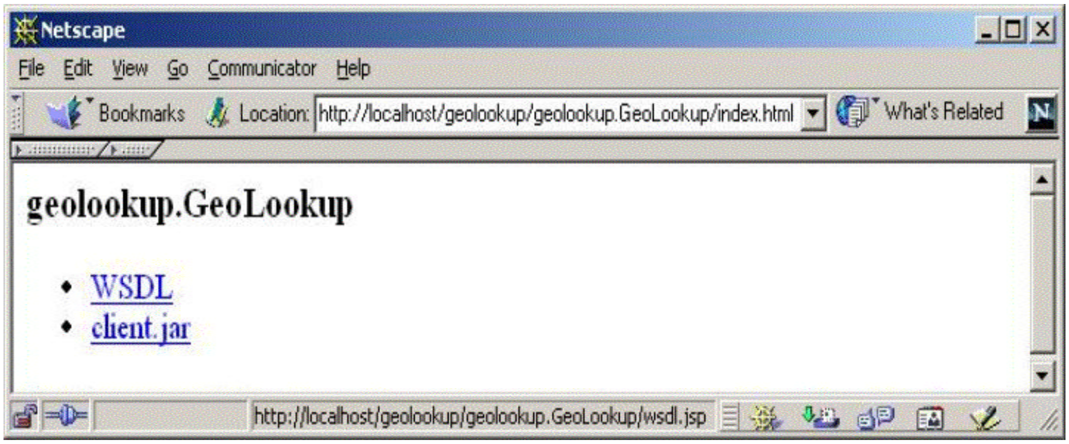
awaiting input

Document: Done (80.595 secs)

(Figure: Geo-Lookup Management)

An administrator manages the GeoLookup service from an administrator web application. Once activated, queries are periodically executed to keep the codes up-to-date. This management console can also be used to lookup particular Lat/Lon values for particular codes.

6.2.1.1 Geo-Lookup Web Service



http://localhost/geolookup/geolookup.GeoLookup/index.html

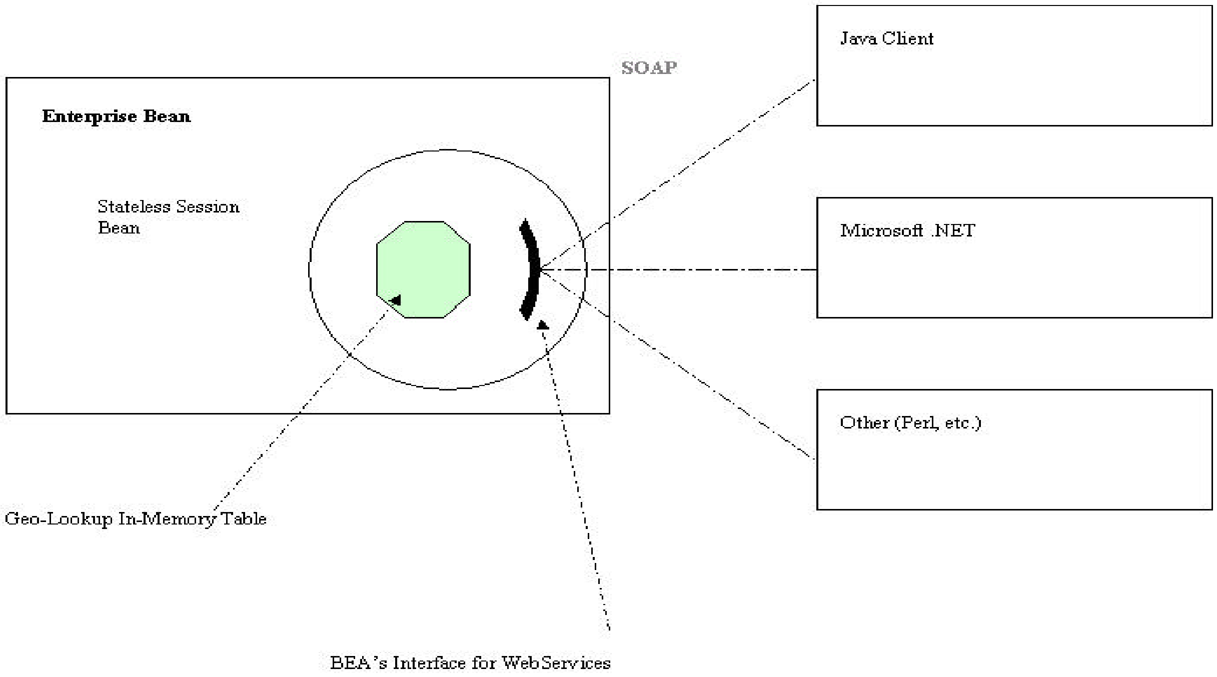
geolookup.GeoLookup

- WSDL
- client.jar

http://localhost/geolookup/geolookup.GeoLookup/wsdl.jsp

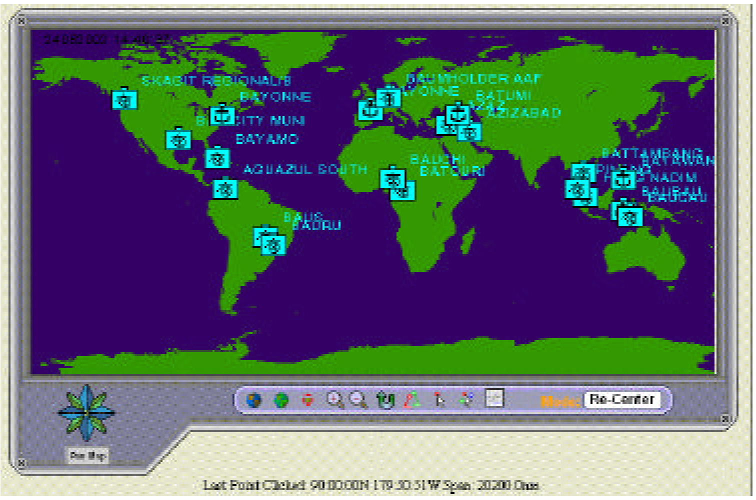
(Figure: GeoLookup Web Service)

Geo-Lookup capability is exposed as a web service to allow an orthogonal scalability across SOAP (Simple Object Access Protocol) consuming clients written in any language. The WSDL (Web Services Definition Language) file exposes an interface to query the GEOLOC service for Lat/Lon values. Java, C++/C# and Perl programmers can connect, therefore, to the service to make use of Lat/Lon values across multiple application tiers. In general, the GeoLookup EJB may transpose into a webservice, which could be accessed from a variety of clients that parse and produce SOAP.



(Figure: GeoLookup EJB Interface)

Currently, the GeoLookup service serves Lat/Lon values that are consumed in the rendering of a WebCOP (Web Common Operating Picture) map. Those same values may appear as variables graphically rendered in a chart or graph on a GCSS web application. As a java-enabled Web Service, the GeoLookup source code and application logic that successfully delivers GEOLOC data to GCSS is not impacted by the nature (or programming language) of its connected client base. Such is the promise of web services.



6.3 Composable FORCENet (CFn)

Composable FORCENet (CFn) is a SPAWAR Systems Center – San Diego (SSC-SD) initiative that started in Code 44 and has been embraced by the Commanding Officer of SSC-SD, and the SSC-SD Executive Director, as an SSC-SD-wide initiative. After a recent demonstration to Admiral Vern Clark, Chief of Naval Operations (CNO), United States Navy, Adm. Clark remarked that Composable FORCEnet was the "best example of a fully netted force that I've ever seen."

Composable FORCEnet is built on the Global Command and Control System, Maritime (GCCS-M) distributed services architecture, and uses as its baseline the actual Program of Record (POR) infrastructure as designed and built by SPAWAR PMW 157. As an SSC prototyping and research initiative, it builds on that solid POR baseline, adding new visualization metaphors, information management techniques, and presentation layers.

It is currently being used and/or extended through multiple projects in multiple codes within SSC-SD and SPAWAR Systems Command, including:

- Unified Command Structure (UCS) (OSD)
- Expeditionary Pervasive Sensing (EPS) (ONR)
- JTF-WARNET
- Collaborative Operations & Responsive Technology Experimentation (CORTEX) (ONR and COMTHIRDFLEET)

Additionally, SSC-SD is working with JFCOM and leveraging the CFn in the Joint Deployable Process Improvement (JDPI) initiative and the Joint Urban Ops (JUO) program. JUO is part of a much larger JFCOM-lead Joint and Coalition initiative dubbed "Multi National Event", and MNE has selected CFn as a distributed architecture for validation in upcoming MNE test events throughout '04.

CFn is composed of GOTS and COTS products. The GOTS products are:

- Navy's Open Source WebCOP – a SPAWAR PMW 157 product
- Geographic Replication Server (GRS) – a SPAWAR PMW 157 product
- Knowledge Web – an SSC prototype that has transitioned as part of the Navy's Collaboration At Sea (CAS) program
- Victor – an ONR knowledge management project built by SSC.

The COTS products include:

- GeoViz – a commercial, 2D/3D mapping and collaboration product
- Other OpenGIS Consortium-compliant applications used as a presentation tier

6.3.1 Web Services in CFn

The CFn initiative is web services based. The WebCOP has within it an Open GIS Consortium (OGC) Web Map Service (WMS), and the GRS is an OGC Web Feature Service

(WFS) with an embedded OGC Web Notification Service (WNS). The GRS is used to store the data used in the common operational picture presented to the users in the WebCOP and GeoViz.

It is also compliant with the RAPIDS guidance for the development of composable, vendor-independent systems.

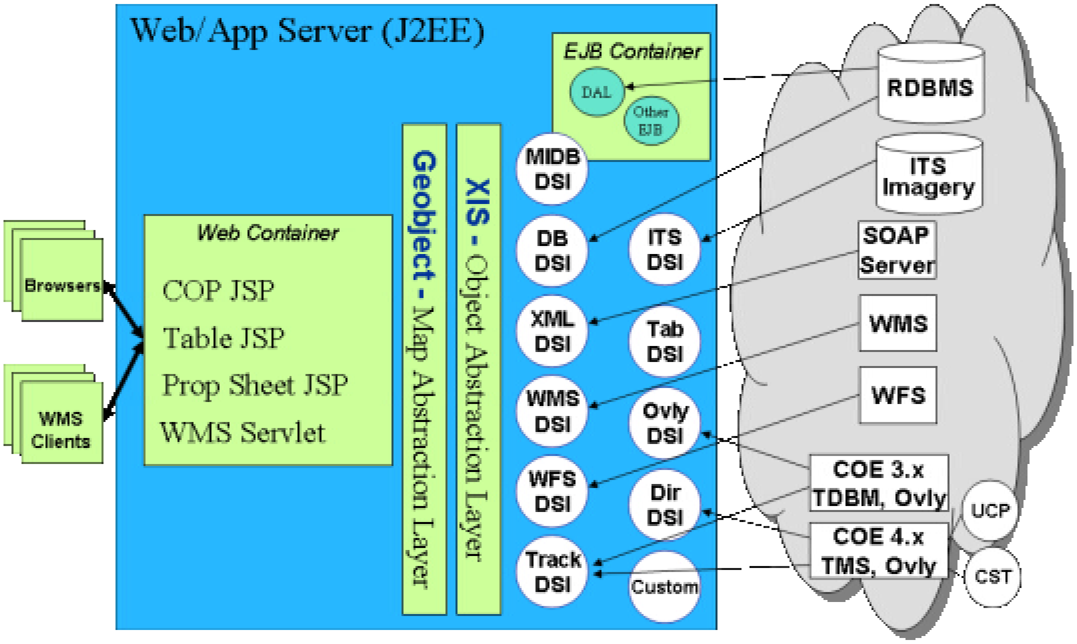
6.3.2 Value to C2 Users

CFn provides an effective capability for integrating data from disparate databases and systems. Portions (GRS, GeoViz, and WebCOP) of CFn were installed on the USS ESSEX for the Integrated Prototype Demonstration (IPD) '03 and were well received. During the exercise, users were able to view the dynamic COP picture from any browser in the battle group. Users could see the GCCS Track picture from more than just the GCCS 3.X terminals throughout the ship. Using the CFn suite, this information (and more) was made available to any browser able to see the CFn servers on board the ESSEX. In addition, 4 PCs were equipped with the GeoViz Internet Explorer Browser PlugIn, enabling users at those workstations to use the highly interactive 2D and 3D displays of the GeoViz tool. GeoViz pulled Track and other C2 data from the CFn servers (the GRS) and displayed them in a 3D dynamic collaborative environment. Users could sit at one GeoViz terminal, annotate their tactical map, and share that map and all of its annotations and data with users at any of the other terminals. Users could also post imagery from any of the GeoViz clients or from any WebCOP browser anywhere in the fleet and have that imagery go directly into the CFn Servers on the ESSEX. From there, that imagery was immediately available to any other WebCOP browser or GeoViz client, without the need for any CONUS interaction or SATCOM hops. Using the Inter-BattleGroup Wireless Network (IBGWN) demonstrated as part of the JTF Warnet exercise, users at any browser in the battle group could instantly collaborate with tactical graphics, imagery, chat, COP data, and maps with the rest of the exercise participants.

6.4 WebCOP Initiatives

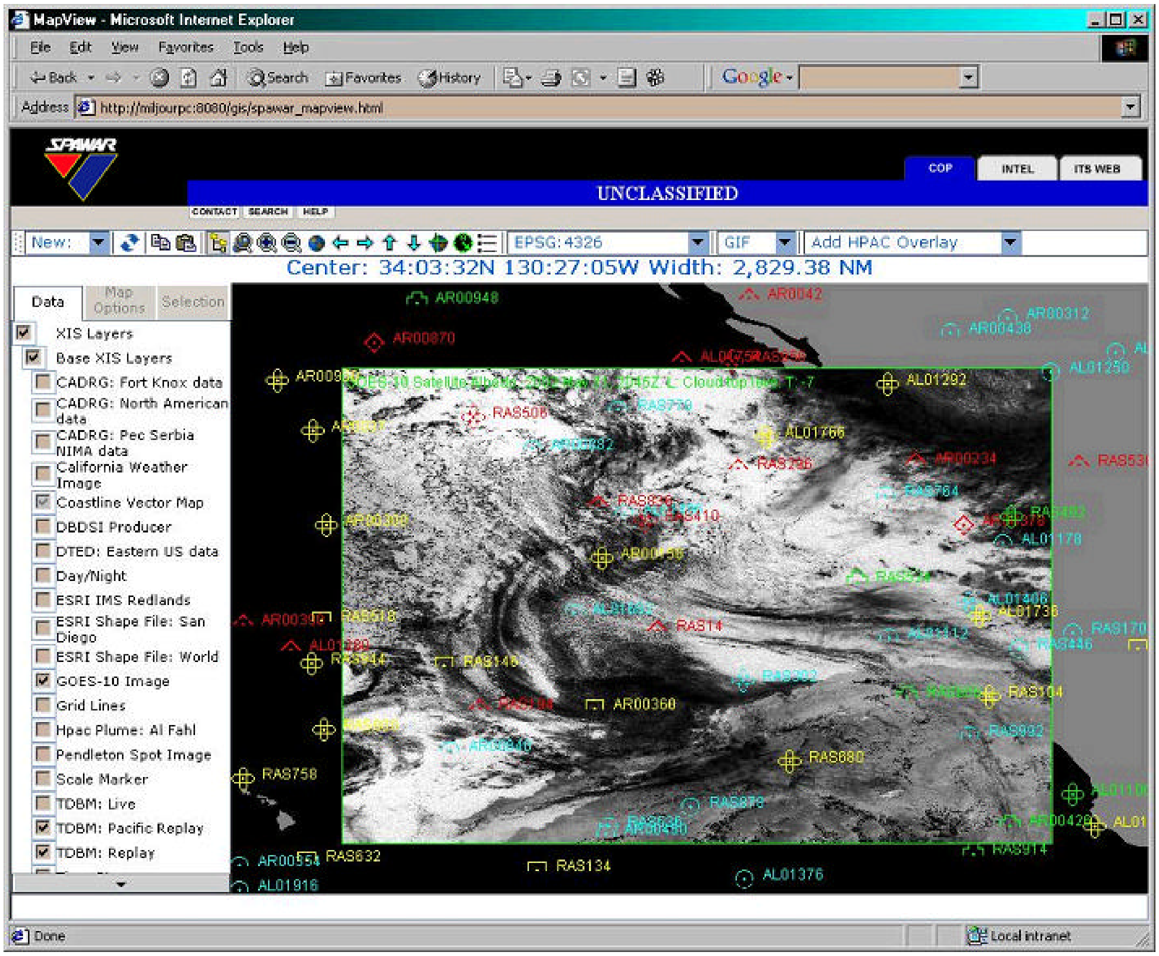
6.4.1 Navy WebCOP

The Navy WebCOP was adopted as one of the components of the GCCS-M and GCCS-I3 systems. It provides interoperability with ITSWeb and IntelShop. The Navy WebCOP provides an Internet browser based solution for viewing a common operational picture (COP). It is capable of displaying TDBM 3.x or 4.x GCCS tracks as well as any other XIS enabled data source such as imagery, Intel data, weather data, overlays, etc. It can also import information from OGC web services using either the WMS or WFS standard, and from other SOAP Web Services. The following figure shows the high level architecture for the Navy WebCOP:



The Navy WebCOP uses HTML, JavaScript and Java Server Pages to generate the situational picture in an Internet Browser. It provides tools to the client for normal map operations such as zooming in, zooming out as well as right click context menus to view information about each track or data item. A property sheet is then displayed with relevant data for the data item(s) that were selected. Under the hood, the WebCOP uses the OGC standard for a WMS server to generate its images in the browser. So in addition to using the WebCOP as a client in a browser, it is also capable of using it in server to server or other custom client solutions that take advantage of the WMS architecture. As mentioned earlier, the Geospatial Replication Service (GRS) can plug into the Navy WebCOP giving it WFS capabilities as well.

The following picture is a screenshot of the Navy WebCOP. One the left hand side you have a list of all the layers available to the user. Included in these are the TDBM tracks as well as the map imagery. The user has the choice of how to customize what data they would like to see on their COP.



Address: http://miljourpc:8080/gis/spawar_mapview.html

New: | EPSG:4326 | GIF | Add HPAC Overlay

Center: 34:03:32N 130:27:05W Width: 2,829.38 NM

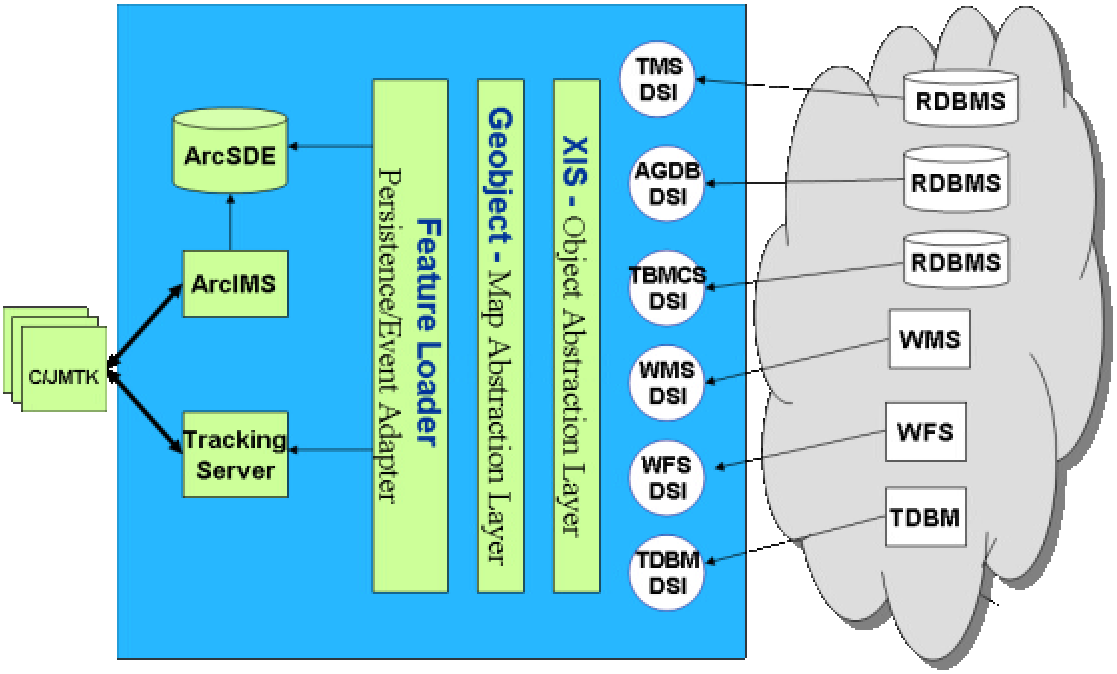
☑ XIS Layers
☑ Base XIS Layers
☐ CADRG: Fort Knox data
☐ CADRG: North American data
☐ CADRG: Pec Serbia NIMA data
☐ California Weather Image
☑ Coastline Vector Map
☐ DBDSI Producer
☐ DTED: Eastern US data
☐ Day/Night
☐ ESRI IMS Redlands
☐ ESRI Shape File: San Diego
☐ ESRI Shape File: World
☑ GOES-10 Image
☐ Grid Lines
☐ Hpac Plume: Al Fahl
☐ Pendleton Spot Image
☐ Scale Marker
☐ TDBM: Live
☑ TDBM: Pacific Replay
☑ TDBM: Replay

Done | Local intranet

6.4.2 Army WebCOP

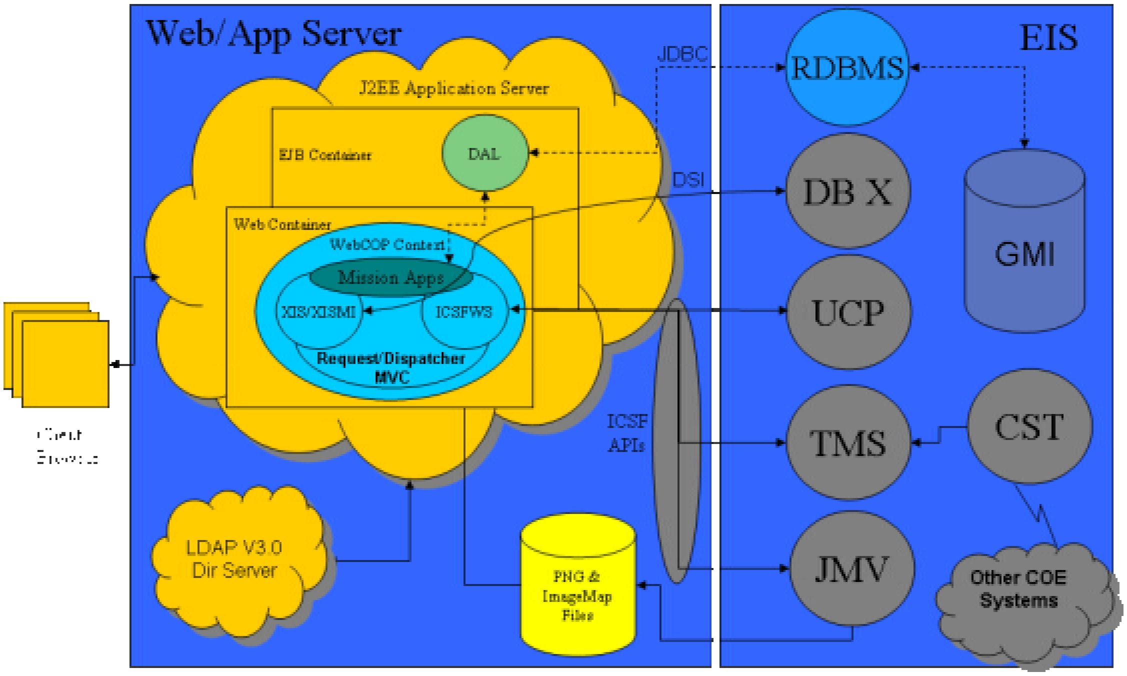
The Army WebCOP provides a similar scope of UI capabilities to the Navy WebCOP, but is built using a very different set of components. It is currently built using CJMTK components from ESRI including ArcSDE, ArcIMS, MapObject Java, ArcXML and MOLE. In addition to these core capabilities the ESRI Tracking Server is expected to provide notification services for clients interested in received near real time updates of data. The army WebCOP, as well as the Navy WebCOP, may be populated with a very diverse set of data sources. In addition to the standard capabilities provided by ArcSDE and ArcIMS, any data source written to the XIS Data Source Interface (DSI) API can be used to populate ArcSDE via the XIS Feature Loader. This capability takes advantage of the already rich set of DSI components that have been developed as part of other efforts, including other WebCOP projects.

The following diagram shows how this capability can be used to collect data from diverse data sources, including OGC compliant WFS and WMS implementations, and standard SOAP Web Services (not shown).

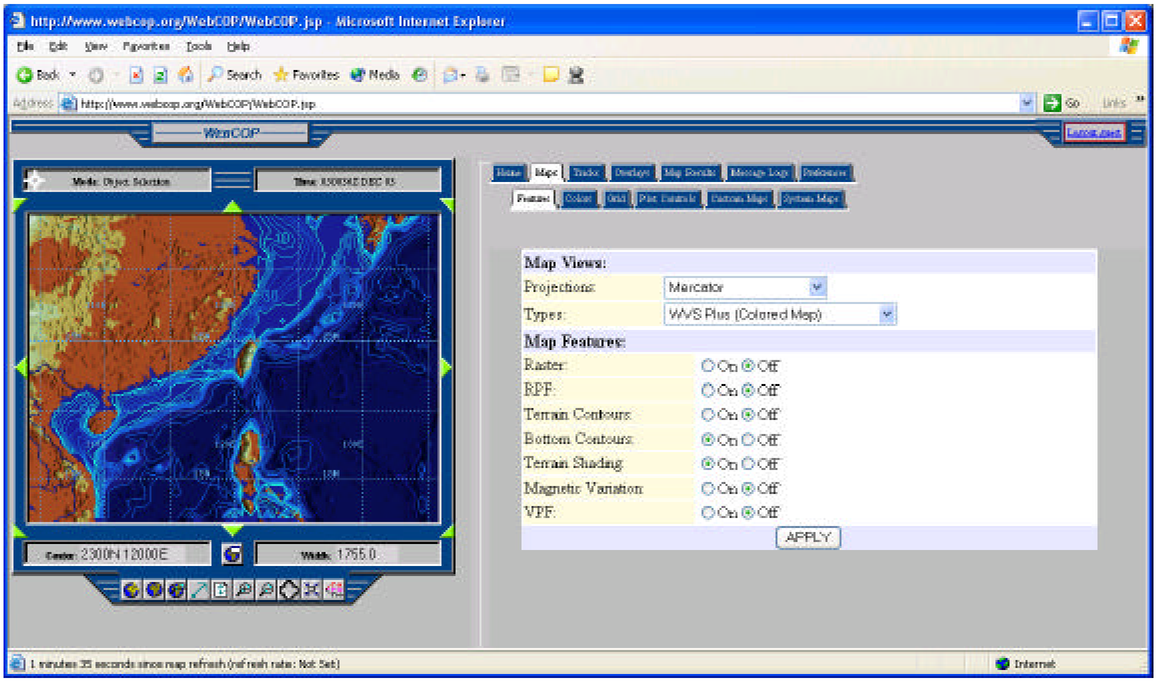


6.4.3 DISA WebCOP

The DISA WebCOP effort is constructed using significantly different APIs than either the Army or Navy WebCOP implementations. Both the DISA WebCOP and the Army WebCOP rely on specialized tools to handle symbology. Where the Army WebCOP uses MOLE for this purpose, the DISA implementation relies on Symplot and JMTK Visualization (JMV), which are part of the COE ICSF (Integrated C4I System Framework) API set. The DISA WebCOP uses LDAP as a directory service for user preferences. There is an XIS extension available for this architecture but this extension has not been widely used, and currently does not support the wide variety of graphic representations currently available to the Navy and Army WebCOP implementations. See the following architecture diagram:



The figure below shows an example of the DISA WebCOP user interface:


http://www.webcop.org/WebCOP/WebCOP.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://www.webcop.org/WebCOP/WebCOP.jsp
Go Links
WebCOP
Logout.east
Mode: Object Selection
Time: 030038Z DEC 03
Home Maps Tracks Overlays Map Records Message Logs Preferences
Features Colors Grid Plot Controls Custom Maps System Maps
Center: 2300N 12000E
Width: 1755.0
1 minutes 35 seconds since map refresh (refresh rate: Not Set)
Internet

Map Views:

Projections: Mercator

Types: WVS Plus (Colored Map)

Map Features:

Raster: ○ On ● Off

RPF: ○ On ● Off

Terrain Contours: ○ On ● Off

Bottom Contours: ● On ○ Off

Terrain Shading: ● On ○ Off

Magnetic Variation: ○ On ● Off

VPF: ○ On ● Off

APPLY

6.4.4 Joint WebCOP

With the similarity of the efforts detailed above it is easy to imagine the amount of potential duplication of effort. Some of the efforts share common implementation schemes, in some cases even allowing them to interchange data sources. The differences between the efforts are important as well. The rendering capabilities are where the components differ the greatest. In the case of the Army WebCOP they have already adopted CJMTK for rendering. The development of a best of bread solution that utilized the best capabilities, tools and architectural choices from the existing solutions is a far more efficient way to create the next generation of applications.

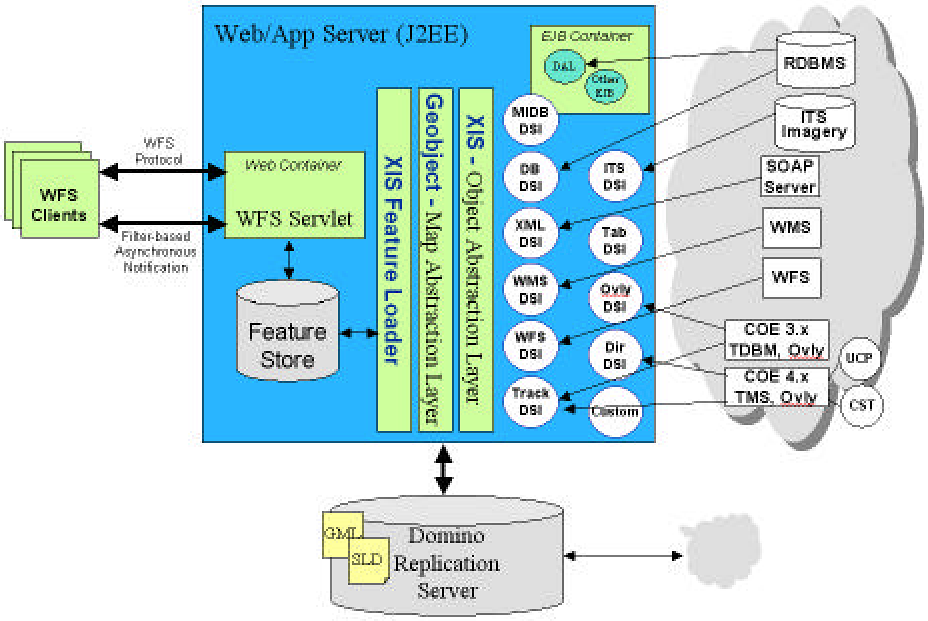
The widely used and well supported CJMTK would be expected to provided the rendering capabilities for any joint effort. The CJMTK will be used in many efforts, and as such there will be more a great deal of expertise in its use within the responsible organizations for the foreseeable future. Other widely used technologies supported by DISA can make important contributions to a joint effort as well, particularly where those technologies are already widely deployed and understood. The capabilities of the existing XIS Feature Loader to import data from new and existing DSI components provides a solid foundation populating the powerful rendering capabilities of the CJMTK with data to make it the most useful COP tool possible.

6.4.5 Geospatial Replication Service

The Geospatial Replication Service (or GRS) was designed to allow clients define a set of data that can be replicated across a network of servers connected in a LAN or WAN configuration. The replicated data could then be accessed from local clients by connecting to its local GRS server. This allows collaboration of data between potentially remote clients.

Each GRS server has a number of clients so it acts as its own local network separate from the rest of the network of GRS servers. By inserting data into a GRS, that data is automatically viewable by other clients connected to the same GRS. If the user chooses to do so, he can share the data with the rest of the GRS network so that users connected to other GRS servers can gain access to the data.

The GRS exposes services to clients using the Web Feature Service (or WFS) interface over HTTP. Essentially it is storage for a set of defined features. A feature can be any set of data of interest that may or may not have a geo-spatial representation for it. Each client connected to a GRS may access the features available to it. Each client may also add to the GRS features it would like to expose to others. The GRS will replicate each set of features available to it with the rest of the GRS network. It does this by using a commercial replication software package called Lotus Domino. The GRS uses SOAP to communicate with the Domino servers. Over time, features are replicated across the GRS network so eventually each GRS will have the same set of data available to them. This gives a C2 user the ability to share data with other remote C2 users and they can make assessments or decisions pertaining to the data that was shared.



Using Domino is optional. The server can run disconnected meaning that the GRS server is running stand alone and only the clients that directly connect to the GRS server can view the data on that server. Another option is to chain WFS servers. Since the GRS is also a WFS server, you can chain one GRS server to another. So a disconnected GRS can expose WFS features that are not contained within that particular GRS. Instead they originate from an external WFS server. So if a client requests features that are external, the GRS server can query the chained WFS server and return the data. This gives a similar effect as replicating the data.

The GRS plugs into the Navy WebCOP so that WebCOP clients can share different layers with others. Those layers will not only be offered up as WFS features, but they will be shared with other GRS servers. From the C2 perspective, this allows C2 users to share part or their entire common operational picture with other C2 users connected to the same GRS network that they are connected to.

6.4.5.1 WFS Request/Response

The following XML code block is a request for the feature type called HPAC where features are contained in the bounding box of 75.0S 39.0E and 65.0S 46.0E.

```xml
<?xml version="1.0"?>
<wfs:GetFeature service="WFS" version="1.0.0"
      xmlns:wfs="http://www.opengis.net/wfs">
  <wfs:Query typeName="HPAC">
    <ogc:Filter>
      <ogc:BBOX>
        <ogc:PropertyName>geoBounds</ogc:PropertyName>
        <gml:Box>
          <gml:coord>
            <gml:X>-75</gml:X>
            <gml:Y>39</gml:Y>
          </gml:coord>
          <gml:coord>
            <gml:X>-65</gml:X>
            <gml:Y>46</gml:Y>
          </gml:coord>
        </gml:Box>
      </ogc:BBOX>
    </ogc:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

The following XML code block is snippet from a WFS response for the above request.

```
<gml:featureMember>
  <hpac:HPACOverlay fid="hpac.0">
    <hpac:name>Fallout Radiation Dose</hpac:name>
    .
    .
    .
    <hpac:eventStartTime>1999-05-16 00:00:00</hpac:eventStartTime>
    <hpac:eventDuration>2635200.0</hpac:eventDuration>
    <hpac:showGeoBounds>false</hpac:showGeoBounds>
    <hpac:geoBounds srsName="EPSG:4326">
      <gml:coord><gml:X>-74.00159</gml:X><gml:Y>40.7347</gml:Y></gml:coord>
      <gml:coord><gml:X>-66.8905</gml:X><gml:Y>45.8819</gml:Y></gml:coord>
    </hpac:geoBounds>
    <hpac:layerLevel>1380525202</hpac:layerLevel>
    <hpac:PlumeGeography>
      <gml:MultiGeometry srsName="EPSG:4326">
      <gml:geometryMember>
        <gml:LinearRing srsName="EPSG:4326">
          <gml:coord><gml:X>-66.8905</gml:X><gml:Y>45.8819</gml:Y></gml:coord>
          <gml:coord><gml:X>-66.9450</gml:X><gml:Y>45.8812</gml:Y></gml:coord>
    .
    .
    .
      <sld:Geometry>
        <ogc:PropertyName>hpac:PlumeGeography/gml:LinearRing</ogc:PropertyName>
      </sld:Geometry>
      <sld:Fill>
        <sld:SvgParameter name="fill">#00042c</sld:SvgParameter>
        <sld:SvgParameter name="fillopacity">0.6</sld:SvgParameter>
        <sld:SvgParameter name="style-fillPattern">#00042c</sld:SvgParameter>
        <sld:SvgParameter name="style-fillStyle">#00042c</sld:SvgParameter>
      </sld:Fill>
      <sld:Stroke>
        <sld:SvgParameter name="stroke">#00042c</sld:SvgParameter>
        <sld:SvgParameter name="strokeopacity">1</sld:SvgParameter>
        <sld:SvgParameter name="style-visible">true</sld:SvgParameter>
      </sld:Stroke>
    .
    .
    .
```

```
<?xml version="1.0" ?>
  <xisml xmlns:pwfs="http://www.polexis.com/pwfs"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:hpac="http://www.polexis.com/hpac">
  <member name="hpac:HPACOverlay">
    <attribute name="hpac:colorTable"
      descriptor="com.xis.domains.display.DisplayDomain.renderingColorTable"/>
    <attribute name="hpac:eventDuration"
      descriptor="com.xis.domains.temporal.TemporalDomain.eventDuration"/>
    <attribute name="hpac:eventStartTime"
      descriptor="com.xis.domains.temporal.TemporalDomain.eventStartTime"/>
    <attribute name="hpac:geoBounds"
      descriptor="com.xis.domains.geo.GeoDomain.geoBounds"/>
    <attribute name="hpac:layerLevel"
      descriptor="com.xis.domains.display.DisplayDomain.layerLevel"/>
    <attribute name="hpac:name"
      descriptor="com.xis.domains.leif.LeifDomain.name"/>
    <attribute name="hpac:showGeoBounds"
      descriptor="com.xis.domains.geo.GeoDomain.showGeoBounds"/>
    <attribute name="hpac:transparency"
      descriptor="com.xis.domains.drawable.DrawableDomain.transparency"/>
    <attribute name="hpac:units"
      descriptor="com.xis.hpac.HpacOverlayTranslator.units"/>
  </member>
  <member name="hpac:Layer">
    <attribute name="hpac:contourColor"
      descriptor="com.xis.hpac.HpacLayerTranslator.contourColor"/>
    <attribute name="hpac:geoBounds"
      descriptor="com.xis.domains.geo.GeoDomain.geoBounds"/>
    <attribute name="hpac:name"
      descriptor="com.xis.domains.leif.LeifDomain.name"/>
    <attribute name="hpac:scale"
      descriptor="com.xis.hpac.HpacLayerTranslator.scale"/>
    <attribute name="hpac:showGeoBounds"
      descriptor="com.xis.domains.geo.GeoDomain.showGeoBounds"/>
    <attribute name="hpac:units"
      descriptor="com.xis.hpac.HpacLayerTranslator.units"/>
  </member>
</xisml>
```

7 Conclusions and Recommendations

7.1 General Guidance

The DoD is clearly already on a path to full adoption of Web Services. We recommend the JTA be updated regularly to accommodate additional and future web services standards that will provide opportunities for more interoperability and more productivity in the integration and/or development of DoD systems.

The Horizontal Fusion (HF) Portfolio Initiative and Network Centric Enterprise Services (NCES) will leverage web services to allow discovery of and access to the right information at the right time by the right people regardless of mission. Web services are a logical implementation mechanism for these goals because of the ubiquity of HTTP and XML, and now SOAP and WSDL. JC2 will be implemented on top of NCES and use its services to implement net-centric C2 and related capabilities (through Mission Capability Packages—MCPs). As discoveries are made about the usage, pros, and cons of individual tools and software packages, we recommend that these discoveries be documented in some forum to be generally available to all DoD software development efforts.

Reusable Application Integration and Development Standards (RAPIDS) and Task Force Web (TFW) provide guidance for standards-based web component (services and user interfaces) development. SAML and federated identity services (such as the Liberty Alliance or Microsoft Passport) should be investigated as a future evolution of DoD Single Sign-On implementations for web-based applications. We also recommend the DoD adopt the WSRP standard to support better web portal interoperability.

The DoD Metadata Registry and Clearinghouse should be expanded to serve as a stable repository for not only XML Schema documents (one of its primary purposes today), but also to other kinds of data that is at the same level of abstraction as XML Schema documents, such as WSDL and OWL.

7.2 Security

We recommend further research and implementation of Web Services security solutions. SAML, XML Signature, XML Encryption, and related standards (outlined in this document and described in the Web Services Standards Analysis Report) should be investigated and a strategy for implementation and even DoD-wide standardization should be created. Systems and net-centric capabilities with true security will only exist when the developers and integrators are given specific guidance and well-defined solutions, by standing up security services on DoD networks, and providing software development kits and examples of their use.

7.3 Messaging

We recommend the DoD continue ongoing efforts to create XML Schema-based message standards to eventually supersede legacy message formats, and also investigate reliable messaging specifications (e.g. WS-Reliability, ebMS, WS-ReliableMessaging, and WS-Acknowledgement), which are still under development.

7.4 Geospatial and Visualization Requirements

For Geospatial applications of Web Services, we recommend the DoD support and recommend the use of Open GIS Consortium (OGC) standards, and sponsor ongoing efforts to improve and expand the existing standards. These standards have been applied in existing C2 applications (many of them described elsewhere in this document). We recommend the DoD officially adopt these OGC standards as the basis for implementation of web-based geospatial services, and the basis for XML representation of geospatial data.

We recommend the DoD require the use of the OGC standards (e.g., WFS, WMS, and other relevant standards), and avoid the use of the proprietary ESRI APIs and formats, for all web services access to CJMTK. We also recommend the CJMTK acquisition authority ensures that these standards are properly implemented, maintained, and advanced as new versions emerge.

And we recommend that the DoD define an XML adaptation of MIL-STD-2525B that is based on commercial standards such as GML and SLD. A standard schema for symbology would be critical to supporting net-centric C2 computing architectures and system interoperability.

7.5 Recommendations for Further Evaluation

This document included a handful of case studies—DoD programs currently employing web services technology. However, these programs are limited by the actual application requirements, and an in-depth study into the broader application of web services technologies is not appropriate. The value of actual hands-on development of demonstrations of web services standards cannot be underestimated. Applications like these help to validate the appropriate and inappropriate uses of web services technologies, and also help to drive requirements for better web services standards in the future. We recommend the DoD support further evaluation of promising web services standards without the constraints normally placed on actual systems development.